

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

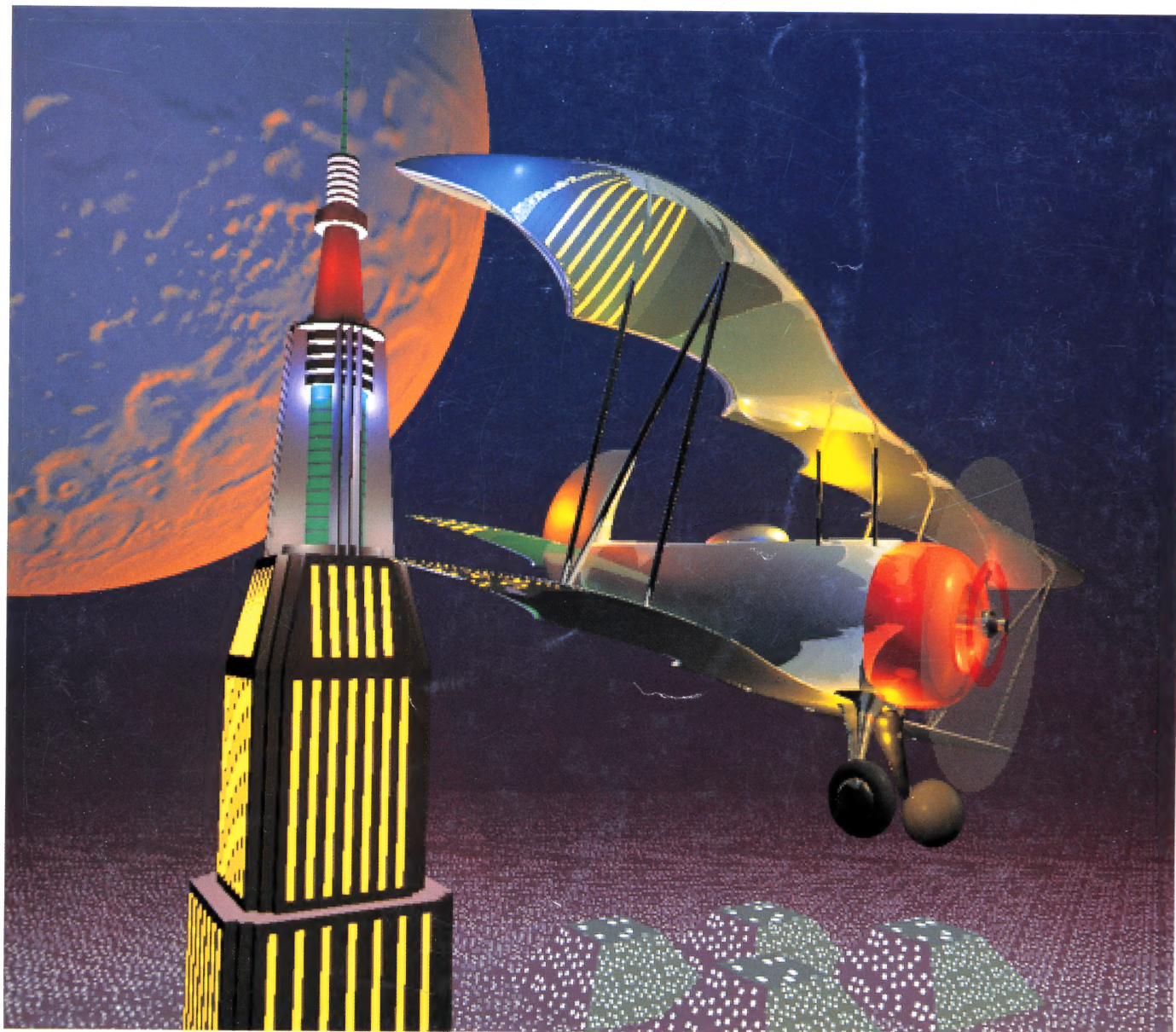
PCX

特集 ADVANCED 2D GRAPHICS

画像回転プログラムXROTO.X
X68000用カードゲームHEART
通巻100号記念特別モニタプレゼント

8
1990

**SOFT
BANK** オーノエックス
定価560円



ひらかれた知性。



サ・ワークステーション。80Mバイトハードディスク、SCSI インターフェイスを標準装備。

SUPER HD

本体+キーボード+マウス+トラックボール

CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

アートの系譜。

EXPERT II

本体+キーボード+マウス+トラックボール

CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンダード。

PRO II

本体+キーボード+マウス

CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)

HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)



次代のユーザーインターフェイスを象徴する“SX-WINDOW”^{*}搭載。

今回のX68000ニューシリーズのデビューに関して、ハードウェア以上にウィンドウ環境の提供に耳目が集中したことは、昨今のビジュアルユーザーインターフェイス事情をふまれば、当然のことと言えるでしょう。マルチウィンドウを駆使してX68000をコントロールする、待ち望まれていた環境がこのSX-WINDOWによって実現されるのです。何の予備知識もなしにこのウィンドウに接した方は、一見して従来のビジュアルシェルのバージョンアップと思われるかもしれませんが、本質的には全く異質のものと云えます。ひとつのウ



ィンドウである仕事をさせながら、別のウィンドウで違う仕事にとりかかる。ひとことで言えばアプリケーションを実行させる環境としてのウィンドウであるということ。これまでのビジュアルシェルではできなかったシーンを生み出しています。複数のアプリケーションを同じ操作のもとで走らせたり、アプリケーション相互でデータのやりとりが可能になるわけです。そして、次代のインテリジェンスを鮮やかに象徴する4階調のハイセンスな画面処理——。SX-WINDOWをターゲットとしたアプリケーション開発もすでに推進されており、これからの展望という点からも大いに期待される場所です。また、このSX-WINDOWはディスクによって供給され、BIOSの高速化(平均2倍)も含めてOSであるHuman68kの機能を拡張。ニューシリーズのみならず、すべてのX68000でこの新しい環境が享受できます。

※SX-WINDOWの起動には、メインメモリ2MBが必要です。CZ-600C/601C/611C/652C/653C/662C/663CでSX-WINDOWをご使用の際には、あらかじめ別売の1MB増設RAMボードを増設してください。

NEW X68000 PERSONAL WORKSTATION SUPER·EXPERT·PRO

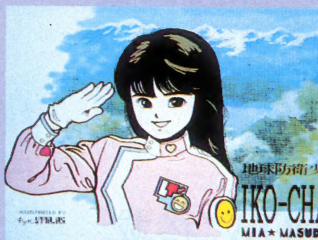
充実のディスプレイラインアップ

15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-602D-BK(ブラック)・GY(グレー)……………標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-605D-BK(ブラック)・GY(グレー)……………標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm)	CZ-613D-TN(チタンブラック)・BK(ブラック)・GY(グレー)……………標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-603D-BK(ブラック)・GY(グレー)……………標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-604D-BK(ブラック)・GY(グレー)……………標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52mm)	CU-21HD-BK(ブラック)……………標準価格148,000円(スピーカー2個同梱・税別)

※印の商品は在庫僅少です。

EXEリーダーズグッズ プレゼント実施中

- いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。
- また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。



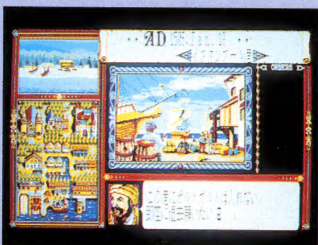
特集 ADVANCED 2D GRAPHICS



カードゲームHEART



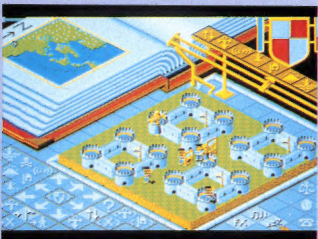
かべくずし



大航海時代



ウルティマV



プロミストランド

Oh!X

C O N T

●特集

40 ADVANCED 2D GRAPHICS

- | | | |
|----|--|------|
| 44 | X68000用グラフィックツール紹介
あなたにあったグラフィックツール | 荻窪 圭 |
| 50 | ギザギザのないグラフィック関数
アンチエイリアシングとは? | 丹 明彦 |
| 68 | X-BASICによる画像処理
後処理によるジャギーの除去 | 中野修一 |
| 72 | 色数の補間と量子化
グラフィックデータを変換する | 鈴木康弘 |
| 77 | 4096色→8色変換
Zの画像をX1で | 亀田雅彦 |

●Oh!X通巻100号記念特別企画

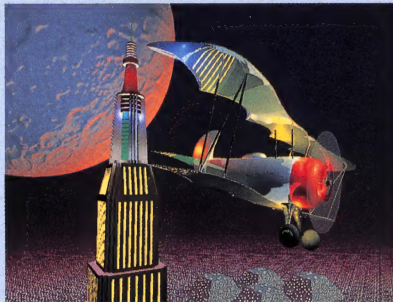
- | | | |
|-----|-------------------|------|
| 23 | 表紙ぎゃらりい | |
| 97 | 対戦ポピュラス 祝一平VS西川善司 | 浦川博之 |
| 100 | 愛読者特大モニタープレゼント | |

●シリーズ全機種共通システム

- | | | |
|-------|-------------------------------------|------|
| 145 | THE SENTINEL | |
| 146 | リンカWLK | 石上達也 |
| ●読みのも | | |
| 158 | 第40回 知能機械概論——お茶目な計算機たち——
人工知能の冒険 | 有田隆也 |
| 160 | 猫とコンピュータ 第50回
サーチャーでござる | 高沢恭子 |
| 162 | X-OVER NIGHT 第3話
旅行あれこれ | 高原秀己 |

<スタッフ>

●編集長/前田 徹 ●編集/植木章夫 岡崎栄子 浅井研二 ●協力/有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本造一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ/杉山和美 ●イラスト/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 AD GREEN ●校正/グループこじら



表紙絵：須藤 牧人

ENTS

●THE SOFTOUCH

28	SOFTWARE INFORMATION 話題のソフトウェア	
32	GAME REVIEW 大航海時代	浦川博之
34	ウルティマ V	荻窪 圭
36	プロミストランド	山田純二
38	AFTER REVIEW 天下統一/ダウントOWN熱血物語 あ〜くしゅ/Yet Another Column	
連載/紹介/講座/プログラム		
81	X68000用画像回転プログラム XROT0.X	渡辺伸也
88	X68000 CARD.FNC用カードゲーム HEART・負けるが勝ち	池谷昌彦
92	X1turbo用ディスク管理プログラムINTEGRAL X1 トランジェントコマンドを作る	亀田雅彦
102	PC-E500テーブルトークRPGサポートシステム(1) ポケコンでCARPGを	松井 信
104	ハードウェア工作入門(2) 基本インタフェイス回路 その2	三沢和彦
107	X-BASICプログラミング調理実習(13) 超入門・ファイル処理	泉 大介
113	X68000マシン語プログラミングChapter_OFH マウスwithグラフィック	村田敏幸
121	PASCALプログラミングへの招待(3) PASCALのデータ型を見る	藤井義巳・藤木健士
126	マシン語カクテル in Z80's Bar 第14回 楽な逆ポーランド?	山田純二
130	(で)のショートプロバてい その12 祝! 1周年記念	古村 聡
134	Oh!X LIVE in '90 OMENS OF LOVE (X68000) ENDLESS RAIN (X1/turbo) ダートフォックスよりRunning up!(X68000MUSICDRVサンプル曲)	小玉和博 伏喜義宏 西川善司

ペンギン情報コーナー.....164
FILES Oh!X.....166
Oh!X質問箱.....168
STUDIO X.....170
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey.....174

1990 AUG. 8

UNIXはAT&T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86, CP/M-68K, CP/M-8000, DR-DOSはDIGITAL RESEARCH
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS CはMICRO SOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事會
WordStar, WordMasterはWORDSTAR International
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハードソンソフト
の商標です。その他、プログラム名、CPUは一般に各メーカーの登録商標です。本文中では"TM", "R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上、PDSと明記されたものを以外、個人で使用するほかの無断複製は禁じられています。

■広告目次

アイソー186
アイビット電子190
アクセス192
アンス・コンサルタンツ9
エスピーエス181
AVCフタバ電機183
オーエーランド187
OKハウス182
計測技研184・185
コナミ12・13
ザイン・ソフト11
J & P表3
システムサコム14・15
シャープ表2・表4・14・8
ソフトクリエイト189
九十九電機22
T & Eソフト17
デンキヤ188
パソコンプラザオクト20・21
P & A18・19
ビクター音楽産業16
満開製作所191(下)
ロゴスシステム10



ディスプレイ関連

カラーディスプレイテレビ



15型カラーディスプレイテレビ
CZ-602D-BK
★CZ-602D-GY
標準価格 99,800円(税別)
(チルトスタンド同梱)

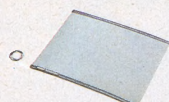


15型カラーディスプレイテレビ
CZ-605D-BK・GY
標準価格 115,000円(税別)
(スピーカー2個・チルトスタンド同梱)



15型カラーディスプレイテレビ
CZ-613D-TN・BK・GY
標準価格 135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

CRTフィルター



高性能CRTフィルター
BF-68PRO
標準価格 19,800円(税別)
(14/15型用)

カラーディスプレイ



14型カラーディスプレイ
CZ-603D-BK・GY
標準価格 84,800円(税別)
(チルトスタンド同梱)



14型カラーディスプレイ
CZ-604D-BK・GY
標準価格 94,800円(税別)
(スピーカー2個・チルトスタンド同梱)



21型カラーディスプレイ
CU-21HD
標準価格 148,000円(税別)
(スピーカー2個同梱)

チューナー



RGBシステムチューナー
CZ-6TU-BK・GY
標準価格 33,100円(税別)
(リモコン付)

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のバラレルデータ伝送を行う場合、別売のスクリーン用バラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。
※2 CZ-603D/604D、CU-21HDをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。
※3 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。

アートツール

画像入力



カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円(税別)



スクリーン用バラレルボード
CZ-6BN1
標準価格 29,800円(税別)

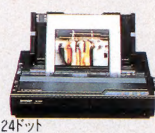
映像入力



カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格 69,800円(税別)

プリンタ

カラープリンタ



24ドット
熱転写カラー漢字プリンタ
★CZ-8PC3
標準価格 65,800円(税別)
(信号ケーブル同梱)



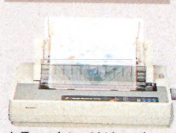
48ドット
熱転写カラー漢字プリンタ
CZ-8PC4
CZ-8PC4-GY
標準価格 99,800円(税別)
(信号ケーブル同梱)

カラービデオプリンタ



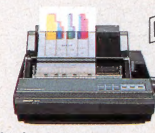
カラービデオプリンタ
CZ-6PV1
標準価格 198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット

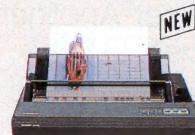


カラーイメージジェット※3
IO-735X
標準価格 248,000円(税別)
(信号ケーブル別売)

ドットプリンタ



24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格 130,000円(税別)
(信号ケーブル同梱)



24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格 160,000円(税別)
(信号ケーブル同梱)



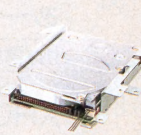
24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格 97,800円(税別)
(信号ケーブル同梱)

ファイル

ハードディスク



ハードディスクユニット(20MB)
CZ-620H
標準価格 178,000円(税別)



増設用ハードディスク
ドライブ(40MB)
(CZ-602C/603C/652C/
653C内蔵用)
CZ-64H
標準価格 120,000円(税別)
(取付費別)

※取付に関してはシャープ
お客様相談窓口にてご
相談ください。

AVturbo シリーズ用 周辺機器

標準価格は税別です。

カラーディスプレイ

●21型カラーディスプレイ※1	CU-21HD	148,000円
-----------------	---------	----------

映像・画像入力編集装置

●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円

●立体映像セット	★CZ-8BR1	29,800円
●パーソナルテロップ※2	CZ-8DT2	44,800円

FM音源

●ステレオタイプFM音源ボード	CZ-8BS1	23,800円
スピーカー(2本1組)標準装備、ミュージックツール同梱		

プリンタ

●24ピンカラー漢字プリンタ(80桁)	CZ-8PG1	130,000円
●24ピンカラー漢字プリンタ(136桁)	CZ-8PG2	160,000円

●24ピン漢字プリンタ(136桁)	CZ-8PK10	97,800円
●24ドット熱転写カラー漢字プリンタ	★CZ-8PC3	65,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4	99,800円
●48ドット熱転写カラー漢字プリンタ	CZ-8PC4-GY	99,800円
●カラービデオプリンタ	CZ-6PV1	198,000円
●カラーイメージジェット	IO-735X	248,000円

ファイル

●ミニフロッピーディスクユニット(2HD・2D)※3	★CZ-520F	118,000円
----------------------------	----------	----------

X68000をサポート。

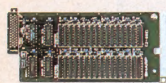


シャープペリフェラルファミリー X68000



ボード

拡張メモリ



1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)



1MB増設RAMボード
(CZ-601C/611C/652C/
653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)



2MB増設RAMボード※4
CZ-6BE2
標準価格 79,800円(税別)



4MB増設RAMボード※4
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス



ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)



GP-1Bボード
CZ-6BG1
標準価格 59,800円(税別)



増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ



数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX



FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI



MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム



モデムユニット※5
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル



RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)



RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード



LANボード
CZ-6BL1
標準価格 268,000円(税別)
CZ-6BL2
標準価格 298,000円(税別)
※電源ユニット/ソフトウェア
(ネットワークドライバVer1.0)同梱

入力



インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)



マウス・トラックボール
CZ-8NM3
標準価格 9,800円(税別)



トラックボール
CZ-8NT1
標準価格 13,800円(税別)



マウス
CZ-8NM2A
標準価格 6,800円(税別)



ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット



拡張I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー



アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック



システムラック
(CZ-600C/601C/602C/603C/
611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

※4 2使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
※5 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・B	5,800円
●高性能CRTフィルター※9	BF-68PRO	19,800円
●スキャナ用パラレルボード※10	CZ-8BN1	27,800円
●品番中の-表示は、B<ブラック>・E<オフィスグレー>を示します。※1 X1ターボシリーズ用 ※2 CZ-862Cには接続できません。※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 I4/I5型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。		

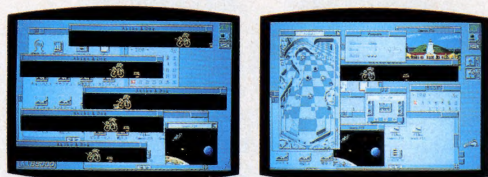
★印の商品は在庫僅少です。

SHARP

"アート"と呼べる高水準のソフトウェアが

(次代のインテリジェンス、 ウィンドウ環境をあなたのX68000で。)

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調+カラー4色、グラフィック面/カラー65,536色中16色)、イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。従来のビジュアルシェルとは異なり、今後のアプリケーションソフトが統一された操作環境で実行できるようになります。



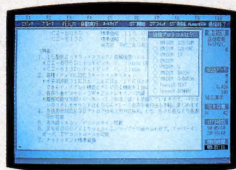
SX-WINDOW ver 1.0

CZ-259SS 10万台達成ご愛用感謝価格6,800円(税別)



(高速通信をサポート。これからの、 そしてさまざまな通信環境に対応する 高機能コミュニケーションソフト。)

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。入出力機能やスクリーンエディタなど豊富な編集機能も魅力です。また、バイナリファイルを転送するプロトコルとしてX modem (128/SUM, 128/CRC, 1K)、Y modem (G, BATCH, G-BATCH)、Transit2 (TEXT, BINARY) プロトコルもサポートしています。



CZ-257CS

標準価格
19,800円(税別)

Communication PRO-68K ver 2.0

(ソースコードデバッグをはじめ、 各種開発ツールを強化。 バージョンアップされたCコンパイラ。)

Cのソースレベルでデバッグできるソースコードデバッグを搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman68k ver 2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(約800種)となっています。強力なMAKEも新たに追加。C言語の標準であるANSI規格準拠をさらに強化し、プロトタイプ宣言もデフォルトに変更されました。「BASIC-Cコンバータ」、「アセンブラ」、「リンク」、「デバッグ」、「ソースコードデバッグ」、「アーカイバ」、「ライブラリアン」、「コンバータ」などのツールが装備されています。



CZ-245LS

7月発売予定

C compiler PRO-68K ver 2.0

X68000をサポート。



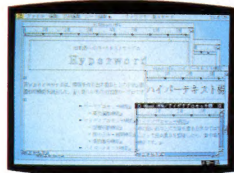
シャープオリジナルソフトウェア
X68000

ビジネスツール

Hyperword

■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。



TOP給与計算エキスパート

■CZ-228BS 標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

TOP財務会計

■CZ-227BS 標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CYBERNOTE PRO-60K

■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。



CARD PRO-60K

■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

CARD PRO-68K用システム手帳リフィル集

■CZ-241BS 標準価格9,800円(税別)

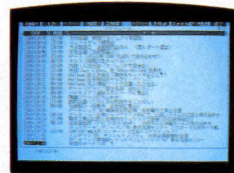
CARD PRO-68K用活用フォーム集

■CZ-242BS 標準価格9,800円(税別)

Stationery PRO-60K

■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。



DATA PRO-60K

■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

BUSINESS PRO-60K

■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

グラフィックライブラリ VOL.1

■CZ-235GS 標準価格8,800円(税別)

着目見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

グラフィックライブラリ VOL.2

■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。



NEW PrintShop PRO-60K

■CZ-221HS 標準価格19,800円(税別)

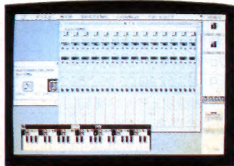
オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。

アートツール

Musicstudio PRO-60K ver.1.1

■CZ-252MS 標準価格28,800円(税別)

24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-68Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。
*MIDIボード(CZ-6BM1)が必要です。



MUSIC PRO-60K (MIDI)

■CZ-247MS 標準価格28,800円(税別)

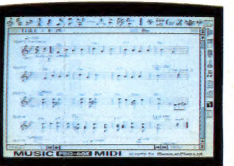
MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI演奏が楽しめます。

*MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ<101曲集>

■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工工作成用からなるライブラリです。



Sampling PRO-60K

■CZ-215MS 標準価格17,800円(税別)

AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-60K

■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-60K

■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ・演奏用ツール。



シューティングゲーム
「ツインビー」
■CZ-217AS 標準価格7,800円(税別)
© KONAMI. 1988



シューティングゲーム
「沙羅曼蛇」
■CZ-218AS 標準価格8,800円(税別)
© KONAMI. 1989



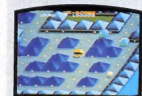
ブロックゲーム
「アルカノイド」
■CZ-222AS 標準価格7,800円(税別)
© TAITO CORP. 1987



ドライブゲーム
「フルスロットル」
■CZ-231AS 標準価格8,800円(税別)
© TAITO CORP. 1988



スポーツゲーム
「熱血高校 ドッジボール部」
■CZ-232AS 標準価格7,800円(税別)
© TECHNOS JAPAN CORP. 1989



アクションゲーム
「バクマニア」
■CZ-233AS 標準価格7,800円(税別)
© NAMCO



アクションゲーム
「ニュージーランドストーリー」
■CZ-230AS 標準価格8,800円(税別)
© TAITO CORP. 1989



スポーツゲーム
「V'BALL」
■CZ-246AS 標準価格7,900円(税別)
© TECHNOS JAPAN CORP. 1989



バイクレーシングゲーム
「スーパーハンガオン」
■CZ-238AS 標準価格8,800円(税別)
© SEGA 1987



ジェットヘリシミュレーションゲーム
「サンダーブレード」
■CZ-239AS 標準価格9,500円(税別)
© SEGA 1987



アクションゲーム
「ダウントウン熱血物語」
■CZ-254AS 標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

OS-9/X68000

■CZ-219SS 標準価格29,800円(税別)

OS-9のもつマルチタスク機能、リアルタイム機能を活かした使いやすい機能的なOS環境を提供。これまでのデータ資産も活かされます。
*OS-9はマイクロウェア社の登録商標です。

Human68k ver.2.0

■CZ-244SS 標準価格9,800円(税別)

THE福袋V2.0

■CZ-224LS 標準価格9,980円(税別)

■AI-68K (Staff LISP/OPS PRO-68K)

■CZ-234LS 標準価格188,000円(税別)

開発ツール

サウンドツール

必聴、必見。

NEWミュージックトレンド“MIDI”体験!!



パソコンミュージック **MIDI**
X68000

音遊サウンドライブ in Summer

● X68000が創造するパソコントレンド、MIDI。

音楽さえ好きであれば、楽器やパソコンが苦手な人でも即エンターティナーになれる、いま注目度一番のニュートレンドです。

● 音遊サウンドライブは、プロのキーボード奏者による本格的なMIDIライブコンサート。

好評の第2弾ではプレイングショーだけでなく、ミュージシャンの楽しいトークや、サウンドスケープ曲あてクイズなど、X68000とMIDIの楽しさを実感して頂けるイベントがぐんと増えました。

● イベント参加者には、オリジナルTシャツやX68000オリジナルグッズをプレゼント。

あなたの感性をとがらせる新鮮で活気あふれるMIDIライブが、あなたをきっと興奮の“音遊”空間へ誘ってくれることでしょう。



EXEクラブが待っている。

● X68000を手にしたら何はともあれ「EXEクラブ」へ。本体同梱の入会申し込みハガキを送るだけで会員証として、オリジナルデザインのカード電卓がもらえちゃう(会費無料)。EXEクラブニュースや最新ソフト、周辺機器などX68000の最新情報を随時ご案内。各種イベント、フェアへのご招待もあります。

(「X68000」は持っているけど、まだ入会していない方も、ぜひこの機会にお申し込み下さい。)

● EXE会員にはEXEリーダーズグッズ・プレゼントも実施中です。詳しくはお近くのEXEショップまで。



NEW X68000、新作ソフト、面白イベント…… まるごと見・体・験フェア。

● 今回のテーマはニューX68000。SUPER-HD/EXPERT II/PRO IIの魅力までにご体験ください。業界注目のSX-WINDOWも必体験。他、新作ソフト体験コーナー、100インチ液晶プロジェクションによる大迫力のゲームたちなど、新しい出会いがあるかもしれません。X68000オリジナルグッズも展示即売。ぜひお近くの会場へお立ち寄りください。

● X68000見体験フェア・音遊サウンドライブ開催日程

開催月日	開催地区	開催場所	お問い合わせTEL
7/20(金)・21(土)	東京	ソフトクリエイティブX68000フェア	03-486-6541 ◎
7/22(日)	太田	パソコンランド21太田店X68000フェア	0276-45-0721 ◎
7/22(日)	金沢	サンミュージックOAプラザX68000フェア	0762-48-6131 ◎
7/22(日)	高松	シャープ見体験フェアイン高松	0878-23-4868 *
7/23(月)	高崎	パソコンランド21高崎東口店X68000フェア	0273-26-5221 ◎
7/28(土)・29(日)	札幌	九十九電機札幌店X68000フェア	011-241-2299 ◎
7/28(土)・29(日)	富山	シャープ見体験フェアイン富山	0762-49-1181 ◎
7/28(土)・29(日)	神戸	星電社三宮本店X68000フェア	078-391-8171 ◎
8/4(土)	高崎	パソコンランド21高崎飯塚店X68000フェア	0273-64-0521 ◎
8/4(土)	京都	J&P京都寺町店X68000フェア	075-341-3571 ◎
8/5(日)	前橋	パソコンランド21前橋店X68000フェア	0272-21-2721 ◎
8/5(日)	姫路	星電社姫路本店X68000フェア	0792-88-1717 ◎
8/5(日)	高知	シャープ見体験フェアイン高知	0888-83-5522 *
8/11(土)・12(日)	宇都宮	計測技研新装開店フェア	0286-22-9811 ◎
8/12(日)	伊勢崎	パソコンランド21伊勢崎店X68000フェア	0270-21-3121 ◎
8/12(日)	東京	T-ZONE X68000フェア	03-257-2650 ◎

◎印の会場では音遊サウンドライブを開催します。＊印の会場には山下肇氏来場。

シャープ株式会社

● お問い合わせは… シャープ(株)電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号(06)621-1221(大代表)

'90 7月27日(金)

新規OPEN!!

CG画像制作センター 秋葉原サテライトオフィス

●新住所

〒101 東京都千代田区外神田6-3-8 外神田田島ビル3F

TEL 03-839-8481 (但し、7月20日より通話可能) (JR秋葉原駅徒歩5分 地下鉄銀座線末広町駅徒歩2分)

——アンス・コンサルタンツ東京本部事務所(現高輪)は7月15日をもって上記へ移転します。——

●オープニング見学会

'90 7月27日(金) 11:00~14:00 お取り引き先、マスコミ、他一般
15:00~20:00 ユーザー様

主な業務案内/CG画像制作プロデュース・アプリケーション開発受託・サイクロンユーザー会ネット
ワークサポート・3次元CAD×CGシステム導入コンサルタント及び教育・アウトプットサービス等々

制作スタッフ募集!! CG画像制作センター

CGプロダクション(仮称:アトリエ68)
として、CG制作ユーザー会・関東支部を開設します。
ユーザーの方はどしどし制作スタッフ登録をして下さい。

※申し込み方法その他詳しくは福岡本社までお問い合わせ下さい

'90 第2回サイクロンCG大会 9月24日に決定!!

全サイクロンシリーズユーザー対象 (98、68、TOWNS)
静止画、アニメその他サイクロンを使用した作品なら何でもOK!!

- 作品受付期間 8月10日~9月8日(当日消印有効)
- 賞金・グランプリ 20万円、その他賞金・賞品多数

※詳細は近日発表

サイクロンExpress α 好評発売中!!

サイクロンExpress α 68.....98,000円
(SHARP X68000)

★CG大会には、 α で応募しよう!!

お知らせ

サイクロンテクニカルセミナー in 大阪

大阪シャープO.A.ショールームにて開催中。
お申し込みはアンスまで。

★ステップ3 7月26日(木)

「ポリゴンを使用する」Z³ TRIPHONY DIGITAL CRAFTとのリンク
.....5,000円

★ステップ4 8月23日(木)

「絵を貼りつける」マッピングの使用法.....5,000円



株式会社アンス・コンサルタンツ

九州本社/〒810 福岡市中央区平丘町68
phone.092-522-6347 FAX.092-521-0400

68000

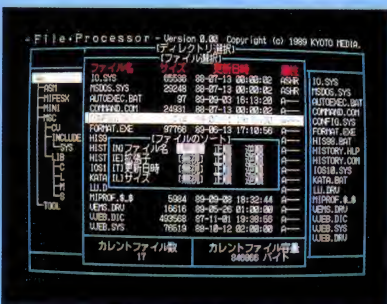
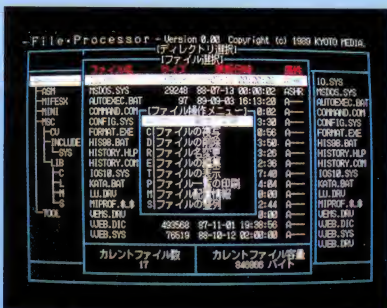
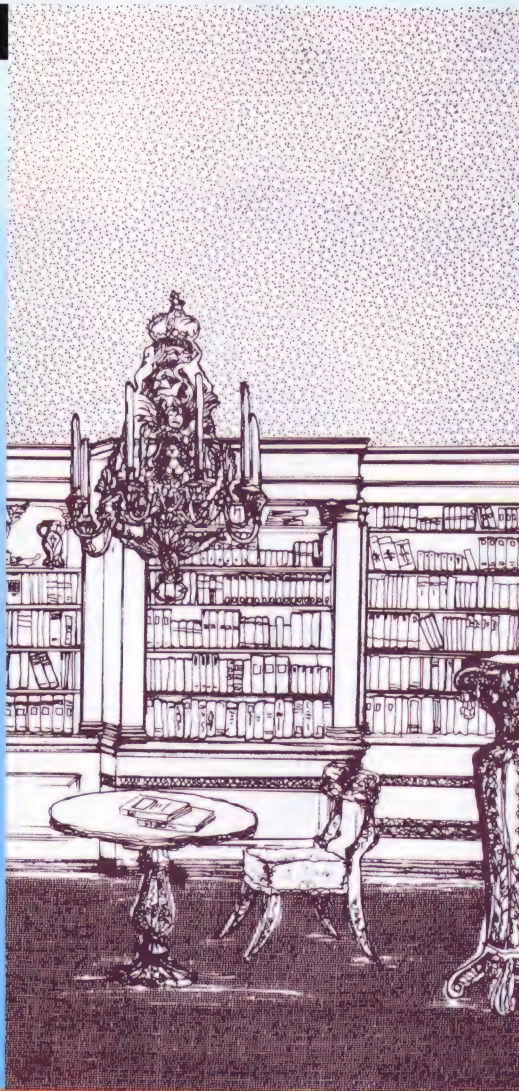
本格的ファイルマネージングソフトウェア

**業界の新星、ロゴスシステムが
ユーザーの希望を1つの形にしました。
これは必要だとか便利じゃない、快感だ!**

全国有名パソコンショップでお求め下さい。
電話1本での通信販売も受付いたしております。

THE FILE PROFESSORの実力

ディスクのバックアップ、ディスクのエディット、ディスクの初期化、ディスクの比較、ディスクの検査、ディスクの情報、FATのエディット、ファイルの検索、ディレクトリのコピー、ディレクトリの削除、ボリュームラベルの設定、ディレクトリの作成、ディレクトリ構造の再読み込み、ディレクトリ構造の印刷、ディレクトリ名の変更、ディレクトリ内容のソート、削除ファイルの復元、ファイル属性の変更、ファイルのコピー/移動、ファイルの削除、ファイルのエディット、ファイルの配置情報、ファイル一覧の印刷、ファイル名の変更、ファイルのソート、ファイル更新日時の変更、ファイルの表示、ファイルの実行、カレンダー、ハードディスクの直撮エディット、システム情報の表示、コマンドシェル、現在時刻の変更。



**メニュー選択方式を実現!!
初心者でも簡単に使える**
(画面写真は、98用を開発中のものです)

ロゴスシステム

このソフトはロゴスシステムのデビュー作です。でも、だからといってなめてもらっちゃあ困ります。私達は、いろいろなソフトを作りました。そのどれもが他社から発売されてきました。出来る事ならば自分達で発売したい/その願いがやっとかないました。

好評発売中!

ロゴスシステム

〒615 京都市右京区西院上今田町17-1 L&Pビル4F
TEL (075) 812-6383 FAX (075) 822-6915

定価 **28,000円**

The File Professor

ザインの彩先端。 ニューコンセプトのアートキャンバス「G=ツール」登場。



「G=ツール」新登場。

新発売
¥28,000

X68000ユーザーのクリエイティブマインドに火をつける新感覚のグラフィックツール。これまでのエディタ概念を払拭し、作品に挑むうえで必要不可欠なグラフィックキャラクター・背景作成のすべてを備えたトータルツールです。ゲームデザインをはじめとしたオリジナルコンピュータアートが驚くほど自由に描けます。今回はグラフィックやスプライトのキャラクターの作成を目的とした「GR EDITモード」をご紹介します。

GR EDITモード

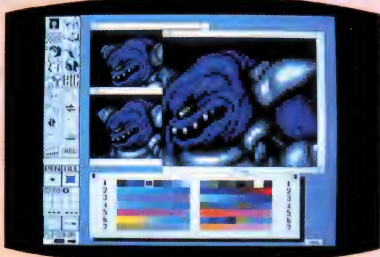


マルチウィンドウシステム:最大12枚まで描画ウィンドウが開ける優れたシステム環境を装備。複数のグラフィック・キャラクターが同時に作成できます。

ユーザーアイコンシステム:パレットやタイル、ペンなど、メインアイコン内の機能を使い勝手に合わせて、自分流のアイコン作成が可能。いちいちポップアップメニューを呼び出す必要もなくアートワークがはかどります。

マウス定義機能システム:マウスの左右クリックボタンに機能定義が可能。たとえば左利きの方もスムーズにオペレーティングできます。

高速メニューウィンドウ処理:メニューウィンドウの開閉は瞬時に。ユーザーアイコンシステムとの併用で、スピーディに仕事が進みます。



zainsoft†

株式会社ザイン・ソフト
〒676兵庫県高砂市米田町米田1162-1
TEL. (0794)31-7453

世にも楽しいシューティングパズル

クオースTM

QUARTH

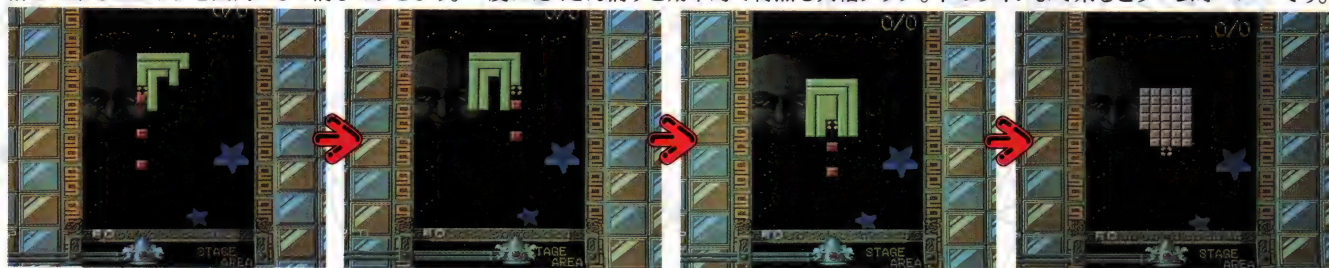
©KONAMI 1990

X68000版
好評発売中
6,800円^(税別)

MSX2版 好評発売中 5,800円^(税別)

PC-9801版 近日発売予定

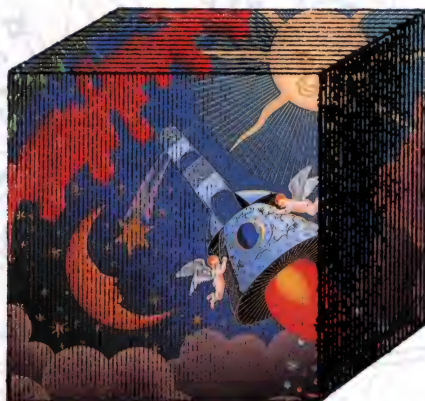
落ちて来るブロックを四角にして消してゆきます。一度にたくさん消すと効率的で得点も大幅アップ。下のラインまで来るとゲームオーバーです。





歴史に残る 前人未踏の 四角い宇宙

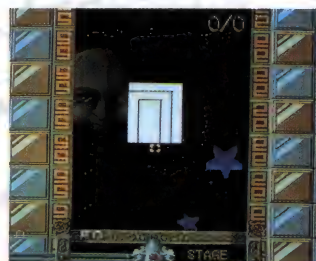
だれもが夢中になれるゲームを創りたい。シンプルでいて奥が深い。だれでも気軽に遊べて、いつまでも飽きない。そんなピュアな、ほんとうの意味でのゲームがしたいと思うことがある。ゲームに対する熱い想いをもう一度じっくりと見つめて今、コナミが新たに発進する、楽園ゲームプロジェクト「クオース」。シューティングの楽しさと、パズルの思考性がマッチングした、すでにゲームセンターでは爆発人気の極楽行き超ソフトだ。ほら、もう引力がココロをズルズルと吸いこんでいる。君も、友も、父も、母も、老若男女を巻きこんで、楽園へ行こう。



協力2人で
仲直りもできる、ますます
熱中の親切設計。



対戦2Pは、
敵と相手の両方と戦う
恐怖のケンカバトルだ。



アイテムブロックが
出るとラッキー。

アーケード版
ジェミニウイング
待望の移植を実現！

ゲームセンターを賑わした
大人気シューティングゲーム
「ジェミニウイングが、
キミのX68Kで今、蘇る！！

MIDI対応

ジェミニウイング Gemini Wing

TM

幾千の流星が降りそそいだ年、世界は轟に覆われていた。人々は孤立し、街は滅び、植物に埋め尽くされた。轟たちはさらに勢いを増し、残された僅かな地さえも蝕んでゆく。そして、ついに最高機密指令第307号、コード名ジェミニウイングは発動された……！

◆特徴◆

- 二人同時プレイ可能
 - MIDI対応(※)
対応楽器 ローランド MT-32
CM-32L CM-64
(※)対応機種ごとに、それぞれ違ったBGMをお楽しみいただけます
(※初期のMT-32では正常に演奏できません)
 - FM音源、ADPCM対応
 - ジョイスティック対応
 - 5.2HD 2枚組
 - X68000 全シリーズ対応
- 標準価格 8,800円

Copyright ©1997/TECMO



闇の血族

美少女名探偵 魅由の繰り広げる

ミステリアスアニメーションアドベンチャー 第1弾!!

THE PREDESTINED HOMICIDES #1

あたし、魅由。

新宿にあるデザイン・スタジオの、新人A・D（アパレル・デザイナー）。……なんだけどあたしの持つて妙な「力」みたいなモノ——人の心が判かっちゃったり、変にカンが良かったり——のせいで、周りからは「名探偵魅由」なんて呼ばれて、よく相談事を持ち込まれたりしている。で、そんなある日、友達のモデルが、突然、殺されてしまった。

そして、あたしの親友だった唯も…… /

これって……ひょっとして連続殺人事件ってヤツ!?



新発売!!

MIDI対応

X68000対応 5"-2HD

●ローランド社MT32完全対応
MIDIインターフェイスボードC-Z-6BIMI
又は、SACOM製SX-68Mが必要です。
(初期のMT-32では、正常に演奏できません。)

標準価格 8,800円



上
巻

伊澤 魅由
(いざわ みゆ)

誕生日:7月16日
身長:168cm
体重:57kg



姫野 里沙
(ひめの りさ)

誕生日:4月2日
身長:163cm
体重:45kg



雪原 リーン
(ゆきはら リーン)

誕生日:2月10日
身長:170cm
体重:53kg



高校生の時、デザイナーの泉麗子に見込まれ、学生生活を営む傍ら麗子のデザインスタジオ（専門学校）に通い始める。そこで小品の手伝いなどをしながら、デザイナーとして本格的に勉強を開始。2年間の研修期間を終え、高校卒業と同時に麗子の強力な推薦で、現在所属している〈スタジオY〇〉に入った。

〈スタジオY〇〉の専属モデル。ファッションショー、雑誌モデルを専門としている。平凡な可愛さがウリで、生活の中で“Y〇（自己性）”をファッションブルに演出する——といった〈スタジオY〇〉のメイン・コンセプトから考えれば、最もY〇らしいモデルと云えるかも知れない。

〈スタジオY〇〉の付属学校、「矢萩デザイナーズ・スタジオ」の卒業生。研修期間中「Y〇プロデュース」でスタイリスト補助のアルバイトをしていた。現在では、Y〇でファッションショーを中心とした若手スタイリストとして活躍中。

東芝EMIより
「38万キロの虚空」CD

新発売 MT税込価格 2,250円
CD税込価格 2,530円

ノベルウェアシリーズ
「38万キロの虚空」

PC-9801 X68000 FM-TOWNS
各9,800円

好評発売中!!
メタルサイト
×68000 8,800円

※標準価格には消費税は含まれておりません。



株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ハードウェア部 03(635)5145
ソフトウェア部 03(635)7609



RPGの概念を一変させた傑作!

1988年発売と同時に世界中のゲーム・フリークを熱狂させた、あの「ダンジョン・マスター」が今、日本中を荒しまわる。

3Dグラフィックスによる複雑な迷路、数々の謎、パーティーを突然襲って来るモンスター。

そしてなによりもプレイヤーの考えること、見ること、手にすること、
すべてにリアルタイムで動いていく……本当の意味のリアルRPGだ。

なぜ、世界をそして日本をこれ程までに興奮させたのか!

その答えは君自身で出して欲しい。

Dungeon Master

ダンジョン・マスター

Master



※画面写真はX-68000版



好評発売中

■X68000
マウス対応

■PC-9801VM21/11, VX, RX, RS, RA ■PC-9800

■PC-9801UV21/11, UX, CV, EX, ES 要バス・マウス/アナログRGB対応

各¥9,800(税抜)

Produced by FTL Games © 1987, 1990 Software Heaven, Inc. © 1990 VICTOR MUSICAL INDUSTRIES, INC.



もう逃げられない!

これが進化した麻雀ソフト、待望のX-68対応発売。

雀豪2

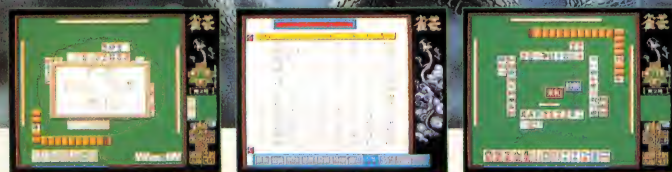
強知能版

麻雀ソフトの決定版登場! プレイすればするほど個性をもったプレイヤーに成長する自己成長型サンプリング機能と、より強化された推論型人工知能の搭載で限りなく実戦麻雀に近づいた。
リアルな4人囲みと見やすい麻雀牌、迫力ある効果音などの採用が麻雀ソフトの金字塔の地位を不動のものにする。

■8月上旬発売:X-68000 ■好評発売中:PC-9801シリーズ

各¥9,800(税抜き)

※画面写真はX-68版の開発画面です



■発売 ビクター音楽産業株式会社

通信販売

当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業株(通信販売係)

X68000

X68000の本質。「黒衣の貴公子」が今解き明かす。



Rune Worth is the world of the boundary between lightness and darkness.
Everything had been born there and then flourished and died there.

©1990 T & E SOFT

黒衣の貴公子



X68000版
7/13 FRI 新発売

neXt

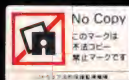
RPG・ACT・SLG、最強のラインナップで
次世代体験…… neXt!

- X68000 5" 2HD 3枚組
- グラフィック書き起こし(高解像グラフィック 512×512ドット)
- ジョイスティック対応
- FM音源8音+ADPCM音源対応
- PC-9801 VM、UVシリーズ PC-286、386シリーズ、NOTE対応
5" 2HD/3.5" 2HD 2枚組 ● サウンドボード対応 ● ジョイスティック対応
- ※ VM、UVはRAM容量640Kバイト以上必要です。
- ※ PC-9801/E/F/M/VF/U2/XAでは、ドライバ、RAM等の増設の如何にかかわらず、作動いたしません。
- ※ また1ドライブのみ搭載のPC-286/LF/LFおよびPC-286/NOTE Executiveでは、
ドライバを複数設置しても作動いたしません。
- ※ 高解像度(640×480ドット)カラーディスプレイをお使いください。液晶ディスプレイにも対応しています。
- PC-8801 SRシリーズ・VA、9800対応 5" 2D 5枚組
- サウンドボード11完全対応、ADPCMをフルサポート ● ジョイスティック対応(9800を除く)
- NEC純正128KRAMボード、I/Oデータ機器製RAMボードに対応したキャッシュドライバ搭載
- MSX2 MSX2+ (RAM64K以上、VRAM128K以上) 3.5" 2DD 3枚組
- MSX-MUSIC(FM音源)対応 ● ジョイスティック対応

標準 各¥8,800

※ 表示価格に消費税は含みません

RPG-neXt……… ルーンワース 黒衣の貴公子
ACT-neXt……… 幻 獣 鬼
SLG-neXt……… 遙かなるオーガスタ



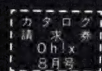
■ 通信販売ご希望の方は現金書留で料金と商品名・機種名と電話番号を明記の上、当社宛お送りください。(遠慮希望の方は300円プラス)
■ カタログご希望の方は、送料として切手200円分を同封の上、カタログ請求券をお送りください。(業者での請求はお断りします)
● T & Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

T&E SOFT

企画・開発・製造・販売

株式会社 ティーアンドイーソフト

〒465 名古屋市中東区豊が丘1810番地 PHONE: 052-773-7770



注目!!冬のボーナス一括払い
手数料(金利)無料

(平成2年12月末支払いをご利用下さい。)

モデム(AIWA) 50台限定 (送料¥1,000)
PV-A24MNP5 (定価 ¥54,800)
 ●MNPクラス5
 ●2400bps
限定特価 ¥26,500
 (送料・消費税込 ¥28,325)

CYBER STICK

●CZ-8NJ2
 (定価 ¥23,800)
超特価!!

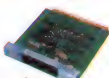
¥18,500 (送料・消費税込み ¥19,570)



X68000シリーズ専用 特価 ¥16,480

MIDIインターフェースボード
SX-68M (サコム)
 (純生コンパチ) 定価 ¥19,800

送料・消費税込み!!



またまた

秋葉原でおなじみの**7/15~8/15**

- お近くの方はお
- 本体単品で特
- ビジネスソフト定

X-1ターボZ III 特別ご提供品!!

台数限定

●CZ-888C+CZ-860D+M-2HD (10枚)
 定価 ¥269,600 ▶ **特価 ¥164,800**

(ボーナス併用も有りますTEL下さい)

・ジョイカード
 ・ゲーム3種
 ・パソコンラック(A)3段
プレゼント中
 送料消費税込み!!

12回	14,400	24回	7,600	36回	5,300	48回	4,100	60回	3,400
-----	--------	-----	-------	-----	-------	-----	-------	-----	-------

ジョイスティック 送料 ¥500

- X-1PRO 定価 ¥9,500 ▶ **特価 ¥7,800**
- ASCII STICK 定価 ¥6,800 ▶ **特価 ¥5,500**

NEW X68000EXPERT II/II-HD & PROII/PROII-HD & SUPER-HD (送料・消費税込)**EXPERT II**

セットでお買い上げの方に、

- ディスク10枚
 - ゲーム3種
 - ジョイカード2枚
- プレゼント中!!

EXPERT II-HD

セットでお買い上げの方に、

- ディスク10枚
 - ゲーム3種
 - ジョイカード2枚
- プレゼント中!!

PROII

セットでお買い上げの方に、

- ディスク10枚
 - ゲーム3種
 - ジョイカード2枚
- プレゼント中!!

PROII-HD

セットでお買い上げの方に、

- ディスク10枚
 - ゲーム3種
 - ジョイカード2枚
- プレゼント中!!

SUPER-HD

セットでお買い上げの方に、

- ディスク10枚
 - ゲーム3種
 - ジョイカード2枚
- プレゼント中!!

EXPERT II

A)セット: CZ-603C+CZ-604D									
12回	?	24回	?	36回	?	48回	?	60回	?
B)セット: CZ-603C+CZ-605D									
12回	30,200	24回	15,900	36回	11,000	48回	8,500	60回	7,100
C)セット: CZ-603C+CZ-613D									
12回	?	24回	?	36回	?	48回	?	60回	?
D)セット: CZ-603C+CU-21HD									
12回	?	24回	?	36回	?	48回	?	60回	?

EXPERT II-HD

A)セット: CZ-613C+CZ-604D									
12回	?	24回	?	36回	?	48回	?	60回	?
B)セット: CZ-613C+CZ-605D									
12回	37,700	24回	19,800	36回	13,700	48回	10,600	60回	8,900
C)セット: CZ-613C+CZ-613D									
12回	?	24回	?	36回	?	48回	?	60回	?
D)セット: CZ-613C+CU-21HD									
12回	?	24回	?	36回	?	48回	?	60回	?

PROII

A)セット: CZ-653C+CZ-604D									
12回	?	24回	?	36回	?	48回	?	60回	?
B)セット: CZ-653C+CZ-605D									
12回	26,800	24回	14,100	36回	9,700	48回	7,600	60回	6,300
C)セット: CZ-653C+CZ-613D									
12回	?	24回	?	36回	?	48回	?	60回	?
D)セット: CZ-653C+CU-21HD									
12回	?	24回	?	36回	?	48回	?	60回	?

PROII-HD

A)セット: CZ-663C+CZ-604D									
12回	?	24回	?	36回	?	48回	?	60回	?
B)セット: CZ-663C+CZ-605D									
12回	34,100	24回	17,900	36回	12,400	48回	9,600	60回	8,100
C)セット: CZ-663C+CZ-613D									
12回	?	24回	?	36回	?	48回	?	60回	?
D)セット: CZ-663C+CU-21HD									
12回	?	24回	?	36回	?	48回	?	60回	?

SUPER-HD

A)セット: CZ-623TN+CZ-604D									
12回	?	24回	?	36回	?	48回	?	60回	?
B)セット: CZ-623TN+CZ-605D									
12回	?	24回	?	36回	?	48回	?	60回	?
C)セット: CZ-623TN+CZ-613D									
12回	42,700	24回	22,500	36回	15,500	48回	12,100	60回	10,100
D)セット: CZ-623TN+CU-21HD									
12回	?	24回	?	36回	?	48回	?	60回	?

X68000シリーズ ~P&Aスペシャルセット=限定誌上販売!!

台数限定 送料・消費税込み

EXPERTセットでお買い上げの方に、
●ディスク10枚 ●ゲーム3種 ●ジョイカード2個 プレゼント中

- CZ-602C+CZ-612D 定価 ¥475,800 ▶ **特価 ¥306,000**
- CZ-602C+CZ-604D 定価 ¥450,800 ▶ **特価 ¥300,000**
- CZ-602C+CZ-605D 定価 ¥471,000 ▶ **特価 ¥320,000**
- CZ-602C+CZ-613D 定価 ¥491,000 ▶ **特価 ¥336,000**
- CZ-602C+CU-21HD 定価 ¥504,000 ▶ **特価 ¥338,000**

EXPERT-HD

- CZ-612C+CZ-612D 定価 ¥585,800 ▶ **特価 ¥375,000**
- CZ-612C+CZ-604D 定価 ¥560,800 ▶ **特価 ¥369,000**
- CZ-612C+CZ-605D 定価 ¥581,000 ▶ **特価 ¥386,000**
- CZ-612C+CZ-613D 定価 ¥601,000 ▶ **特価 ¥403,000**
- CZ-612C+CU-21HD 定価 ¥614,000 ▶ **特価 ¥407,000**

PRO-HD

- CZ-662C+CZ-612D 定価 ¥527,800 ▶ **特価 ¥339,000**
- CZ-662C+CZ-604D 定価 ¥502,800 ▶ **特価 ¥333,000**
- CZ-662C+CZ-605D 定価 ¥523,000 ▶ **特価 ¥352,000**
- CZ-662C+CZ-613D 定価 ¥543,000 ▶ **特価 ¥368,000**
- CZ-662C+CU-21HD 定価 ¥556,000 ▶ **特価 ¥372,000**

1回～84回払いまでOK!!

★頭金なし!★即日発送

P&Aがズバリ超特価セールでご奉仕!!

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

立寄り下さい。専門係員が説明いたします。

価で受付します。詳しくは電話にてお問合せ下さい。

価の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー (送料1ヶ～5ヶまで¥500)

Z's STAFF PRO68K Ver2.0 (ツァイト)	定価 ¥ 58,000	特価 ¥ 39,700
Z's TRIPHONY デジタルクラフト (ツァイト)	定価 ¥ 39,800	特価 ¥ 29,300
テラッソ (ハミングバード)	定価 ¥ 19,800	特価 ¥ 15,800
KAMIKAZE (サムシング・グッド)	定価 ¥ 68,800	特価 ¥ 46,000
EW & EI (イースト)	定価 ¥ 38,800	特価 ¥ 28,800
C & Professional Pack (マイクロウェアジャパン)	定価 ¥ 58,800	特価 ¥ 43,000
Final Ver 3.2 (エーエスピー)	定価 ¥ 38,000	特価 ¥ 30,000
DATA PRO68K CZ220BS	定価 ¥ 58,000	特価 ¥ P&A特価
CARD PRO68K CZ226BS	定価 ¥ 29,800	TEL下さい
C compiler PRO68K CZ211LS	定価 ¥ 39,800	特価 ¥ 32,000
OS-9/X68000 CZ219SS	定価 ¥ 29,800	特価 ¥ P&A特価 TEL下さい
AI-88K CZ234LS	定価 ¥ 188,000	特価 ¥ 143,000
THE 複製 V2.0 CZ224LS	定価 ¥ 9,800	特価 ¥ 7,700
SOUND PRO68K	定価 ¥ 15,800	特価 ¥ 12,500
MUSIC PRO68K CZ213MS	定価 ¥ 15,800	特価 ¥ P&A特価 TEL下さい
Sampling PRO68K CZ215MS	定価 ¥ 17,800	特価 ¥ 14,000
MUSIC-studio PRO68K 237MS	定価 ¥ 15,800	特価 ¥ P&A特価 TEL下さい
MUSIC-PRO68K (MIDI) 247MS	定価 ¥ 28,800	特価 ¥ 22,000
New-print Shop CZ11HS	定価 ¥ 19,800	特価 ¥ P&A特価
Communication 23GS	定価 ¥ 19,800	TEL下さい
C-TRACE68 Ver.3.0 (キャスト)	定価 ¥ 98,000	特価 ¥ 77,000
サイクロン EXPRESS α68	定価 ¥ 98,000	特価 ¥ 72,000
G68K Ver2 PRO	定価 ¥ 22,000	特価 ¥ 16,300
THE FILE PROFESSOR (ロゴシステム)	定価 ¥ 28,000	特価 ¥ 20,500
Qツール (ザインソフト)	定価 ¥ 28,000	特価 ¥ 20,500
ターミネーター2 (SPS)	定価 ¥ 17,800	特価 ¥ 13,500
マジックバレット (ミュージカルプラン)	定価 ¥ 19,800	特価 ¥ 14,900
Hyper word CZ-251BS	定価 ¥ 39,800	特価 ¥ 30,900

●ゲームソフト20%OFF OK!! (一部ソフト除く)

周辺機器コーナー (送料 ¥1,000)

A) CZ-8NSI	定価 ¥ 188,000	特価 ¥ 145,000
B) CZ-6VTI	定価 ¥ 69,800	特価 ¥ 54,000
C) CZ-6TU	定価 ¥ 33,100	特価 ¥ 25,000
D) BF-68PRO	定価 ¥ 19,800	特価 ¥ 15,500
E) CZ-6BEI	定価 ¥ 35,000	特価 ¥ 26,500
F) CZ-6BEIA	定価 ¥ 38,000	特価 ¥ 28,600
G) CZ-6BE2	定価 ¥ 79,800	特価 ¥ 60,000
H) CZ-6BE4	定価 ¥ 138,000	特価 ¥ 107,000
I) CZ-6BF1	定価 ¥ 49,800	特価 ¥ 38,200
J) CZ-6BPI	定価 ¥ 79,800	特価 ¥ 61,000
K) CZ-6BBI	定価 ¥ 26,800	特価 ¥ 20,300
L) CZ-6EBI	定価 ¥ 88,000	特価 ¥ 67,500
M) AN-S100	定価 ¥ 36,600	特価 ¥ 28,500
N) CZ-6SDI	定価 ¥ 44,800	特価 ¥ 35,000
O) CZ-8PC3	定価 ¥ 65,800	
P) CZ-8PC4	定価 ¥ 99,800	
Q) CZ-8PG1	定価 ¥ 130,000	
R) CZ-8PG2	定価 ¥ 160,000	
S) CZ-8PK10	定価 ¥ 97,800	
T) CZ-6PVI	定価 ¥ 198,000	特価 ¥ 153,000
U) IO-735X	定価 ¥ 248,000	特価 ¥ 190,000
V) CZ-8BSI	定価 ¥ 23,800	特価 ¥ 19,000

W) PIO-6BE1-A (I/O DATA) (X) PIO-6BE2-2M (I/O DATA) (Y) PIO-6BE4-4M (I/O DATA)
定価 ¥ 25,000 特価 ¥ 18,200 定価 ¥ 50,000 特価 ¥ 36,800 定価 ¥ 88,000 特価 ¥ 64,800

中古パソコンはP&Aにお任せ!!

その場で高価現金買取り・高価下取りOK!!

- まずはお電話下さい。下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。
- 下取りの場合 価格は常に変動しますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)
- 買取りの場合 現品が着次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。
- 近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

- 月々¥1,000円からOK!! ●ボーナス払いOK (夏冬10回までOK)
- 支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全

全商品保証付。専門の担当者がお客様の立場で対応します。初期不良、輸送トラブル etc. 万が一初期不良、輸送トラブルが発生した際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

- マイコン
- ビデオ
- ビデオテープ

P&A

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号

03-651-0148 (代) 03-651-0141

営業時間
平日: AM10:00～PM7:00
日祭: AM10:00～PM6:00



超特価でクレジットが組める!!

X68000用ハードディスク (送料 ¥1,000)

アイテム

- HXD-040 (40MB/23ms) 定価 ¥118,000 特価 ¥ 88,000
- HXD-042 (増設用) 定価 ¥128,000 特価 ¥ 95,000

アイテム

- ITX-640 (40MB/28ms) 定価 ¥158,000 特価 ¥101,000
- ITX-680 (80MB/20ms) 定価 ¥198,000 特価 ¥131,000

プリンター (ケーブル・用紙付) 限定5台 新品 (送料 ¥1,000)

- CZ-8PC3 (カラー漢字24ドット熱転写プリンター)
定価 ¥65,800 特価 ¥39,800
- CZ-8PK8 (24ピン漢字プリンター136桁)
定価 ¥152,000 特価 ¥69,000
- CZ-8PC4 P&A特選!! (カラー漢字48ドット熱転写プリンター)
定価 ¥99,800 特価 ¥56,000

モデムコーナー (送料 ¥1,000)

- (A) MD-24FS5 (オムロン) 定価 ¥ 49,800 特価 ¥ 34,800
- (B) MD-24FS7 (オムロン) 定価 ¥ 64,800 特価 ¥ 45,000
- (C) コムスター2424/4 (NEC) 定価 ¥ 38,800 特価 ¥ 28,000
- (D) コムスター2424/5 (NEC) 定価 ¥ 44,800 特価 ¥ 32,000

P & A 特選パソコンラック (送料 無料) 移動自由 (キャスター付)

③ 3段	④ 4段	⑤ 5段
875 (H)	1320 (H)	1280 (H)
× 580 (D)	× 600 (D)	× 600 (D)
× 610 (W)	× 630 (W)	× 620 (W)
¥9,000	¥12,000	¥15,000

中古パソコン

送料 ¥2,000

- X-68000セット ¥210,000 ●CZ-856C ¥45,000 ●CU-14AG2 ¥30,000
- X-68000ACEセット ¥240,000 ●CZ-870C ¥55,000 ●CU-14H2 ¥30,000
- X-1ターボZセット ¥100,000 ●CZ-881C ¥65,000 ●CZ-8PC2 ¥25,000
- X-1G/30セット ¥39,000 ●CZ-820D ¥10,000 ●CZ-8PK6 ¥32,000
- CZ-822C ¥15,000 ●CU-14GB ¥5,000
- CZ-830C ¥25,000 ●CU-14BD ¥25,000

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕

- 商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕

- 銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様の住所・お名前・商品名等をお知らせください。

(電話扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 (株)ピー・アンド・エー

〔クレジットでお申し込みの方〕

- 電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
- 現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
- 1回～84回払いまで出来ます。但し、1回のお支払いは¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	2.5	3.5	5.0	5.0	9.0	10.5	14.5	19.0	24.5	32.0	38.5

■平成2年夏のボーナス一括払い(8月末)OK!!手数料ナシ!!おトクです。ぜひ!!超低金利クレジットをご利用下さい。

パソコンプラザ



案内図



店頭セール実施中

—'90 オクトで始まるパソコンワールド—

03-730-6271

●営業時間 **AM 11:00 ~ 9:00**/日曜・祭日**PM 7:00** 電話一本で、ハイ即納
〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販 ●定休日毎週火曜日 祭日の場合翌日になります。

オクト
ラクラククレジット


1回	2%	3回	2.5%	6回	3.5%	10回	5%	12回	5%	15回	7.5%
18回	9%	20回	10%	24回	11%	30回	14.5%	36回	15.5%	48回	20%

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回~60回)頭金ナシOK!
- ▶ボーナス一括払いOK! ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリースタイルシステム
- ▶店頭デモンストレーション実施中

オクト
セレクトシステム

広告掲載商品以外の製品も取扱っております。



OCT-1 蒲田 ●平成2年、8月末一括払い(手数料ナシ)OK!!
OKだよ〜。超低金利 ハッピークレジットです!!
EXPERT II・PRO II 新発売記念セール開催中!!

OPEN

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚 ②ジョイカード 2個(連射式) ③シリコンキーボードカバー

お好みのセットをお選び下さい。送料無料

●SX-WINDOW搭載。

●40Mバイトハードディスク搭載



EXPERT II・EXPERT II-HD

- CZ-603C-BK/GY 定価 ¥338,000
- CZ-613C-BK/GY 定価 ¥448,000

現金特価!! 推選
お電話下さい。

●SX-WINDOW搭載。

●拡張I/Oポート4スロット装備



PRO II・PRO II-HD

- CZ-653C-BK/GY 定価 ¥285,000
- CZ-663C-BK/GY 定価 ¥395,000

CZ-8NJ2

●インテリジェントコントローラ

定価 ¥23,800

超特価 ¥18,800





CZ-605D-GY/BK

定価 ¥115,000



CZ-613D-GY/BK

定価 ¥135,000



CZ-604D-GY/BK

定価 ¥94,800



CU-21HD

定価 ¥148,000

① CZ-603C + CZ-605D 定価合計 ¥453,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

② CZ-613C + CZ-605D 定価合計 ¥563,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

③ CZ-653C + CZ-605D 定価合計 ¥400,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

④ CZ-663C + CZ-605D 定価合計 ¥510,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑤ CZ-603C + CZ-613D 定価合計 ¥473,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑥ CZ-613C + CZ-613D 定価合計 ¥583,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑦ CZ-653C + CZ-613D 定価合計 ¥420,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑧ CZ-663C + CZ-613D 定価合計 ¥530,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑨ CZ-603C + CZ-604D 定価合計 ¥429,800 ▶ **オクト大特価**

12回	¥28,000	24回	¥14,800	36回	¥10,200	48回	¥8,000
-----	---------	-----	---------	-----	---------	-----	--------

⑩ CZ-613C + CZ-604D 定価合計 ¥542,000 ▶ **オクト大特価**

12回	¥36,000	24回	¥19,000	36回	¥13,100	48回	¥10,200
-----	---------	-----	---------	-----	---------	-----	---------

⑪ CZ-653C + CZ-604D 定価合計 ¥379,800 ▶ **オクト大特価**

12回	¥25,400	24回	¥13,400	36回	¥9,300	48回	¥7,200
-----	---------	-----	---------	-----	--------	-----	--------

⑫ CZ-663C + CZ-604D 定価合計 ¥489,800 ▶ **オクト大特価**

12回	¥32,200	24回	¥17,000	36回	¥11,800	48回	¥9,200
-----	---------	-----	---------	-----	---------	-----	--------

⑬ CZ-603C + CU-21HD 定価合計 ¥486,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑭ CZ-613C + CU-21HD 定価合計 ¥596,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑮ CZ-653C + CU-21HD 定価合計 ¥433,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

⑯ CZ-663C + CU-21HD 定価合計 ¥543,000 ▶ **オクト大特価**

12回	?	24回	?	36回	?	48回	?
-----	---	-----	---	-----	---	-----	---

♡どんどんTELしよう。安くなるかもヨ!!

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

※クレジットの回数は1回~60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料 ●店頭デモ実施中...専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)・超低金利・ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の
製品も取扱っております。

チャンスノ X68000・SUPER-HD(チタン)= 好評・発売中
どんどんTEL下さいネ。

送料 ¥2,000

X68000 EXPERT-HD

SX-WINDOW搭載。



●ザ・ワークステーションと呼ぶにふさわしい
スーパーな68000!! 新登場!!
SUPER-HD。

※プレゼント① MD-2HD10枚 ③ ジョイカード(連射式)
② アフターバーナー(¥9,200) ④ シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN

定価合計 ¥633,000... 大特価!! TEL下さい。

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回 ? 24回 ? 36回 ? 48回 ?

♡安くてゴメンなさい。今だけヨ!!

他のディスプレイ 1 CZ-602D, 2 612D, 3 CZ-603D,
4 CU-21HDの組合せもごさいますのでお問い合わせ
下さい。

※超低金利クレジットご利用下さい。1回~60回払い、頭金ナシノボーナス1回払い、ボーナス2回払いOK!!

オクト限定スペシャルセット



ラストチャンス!!
早い者勝ち!!

●CZ-612C(BK)
(¥466,000)
●CZ-602D(BK)
(¥99,800)
●MD-2HD 10枚
●ジョイカード(連射式×2個)
●ゲーム

オクト超特価

¥364,000(送料・消費税込み!!)

※ディスプレイ=①CZ-604D ②CZ-605D
③CZ-613D ④CU-21HD
との組合せもごさいます。TEL下さい。

オクト特選 シャープ周辺機器 (送料 ¥1,000)

●CZ-6BE1 IBM増設RAMボード.....(¥35,000)▶大特価 ¥26,500
●CZ-6BE1B IMB増設RAMボード.....(¥28,000)▶大特価 ¥21,000
●CZ-6BE2 2MB増設RAMボード.....(¥79,800)▶大特価 ¥60,500
●CZ-6BE4 4MB増設RAMボード.....(¥138,000)▶大特価 ¥104,800
●CZ-6BF1 増設用RS-232Cボード.....(¥49,800)▶大特価 ¥38,500
●CZ-6BG1 GP4Bボード.....(¥59,800)▶大特価 ¥45,000
●CZ-6BM1 MIDIボード.....(¥26,800)▶大特価 ¥20,500
●CZ-6BN1 スキャナ用ハラルボード.....(¥29,800)▶大特価 ¥22,800
●CZ-6BP1 数値演算プロセッサボード.....(¥79,800)▶大特価 ¥60,500
●CZ-6B01 ユニバーサルI/Oボード.....(¥39,800)▶大特価 ¥30,500
●CZ-6EB1/BK カラーイメージ・ユニット.....(¥88,000)▶大特価 ¥66,800
●CZ-6VT1/BK カラーイメージ・ユニット.....(¥69,800)▶大特価 ¥53,000
●CZ-6BL1 LANボード.....(¥268,000)▶大特価

●CZ-8NM2A マウス.....(¥68,800)▶大特価 ¥5,300
●CZ-8NT1 マウス・トラックボール.....(¥98,800)▶大特価 ¥7,500
●CZ-8NS1 カラーイメージスキャナ.....(¥188,000)▶大特価
●CZ-8BC1 FAXボード.....(¥79,800)▶大特価 ¥60,500
●CZ-8TM2 モデムユニット.....(¥49,800)▶大特価 ¥38,000
●CZ-6H 増設ハードディスク.....(¥120,000)▶大特価
●CZ-6TU GY/BK RGBシステムチューナー.....(¥33,100)▶大特価 ¥25,000
●BF-68PRO 高性能CRTフィルター.....(¥19,800)▶大特価 ¥15,500
●SX-68M(システムサム) MIDIボード.....(¥19,800)▶大特価 ¥15,000
●PIO-68BE1-A(I/O DATA) IMB増設RAMボード.....(¥25,000)▶大特価 ¥18,500
●PIO-6BE2-2M(I/O DATA) 2MB増設RAMボード.....(¥50,000)▶大特価 ¥37,000
●PIO-6BE4-4M(I/O DATA) 3MB増設RAMボード.....(¥88,000)▶大特価 ¥65,000

オクト面白グッズ

アイテック(送料 ¥1,000)

●IT-X640(¥158,000)
.....大特価 ¥103,000
●IT-X680(¥198,000)
.....大特価 ¥134,000

モデムコーナー(送料 ¥1,000)

●MD-I200A III.....大特価 ¥14,800
●MD-24FS4.....大特価 ¥31,500
●MD-24FS5.....大特価 ¥34,800
●MD-24FP4.....大特価 ¥27,900
●MD-12FS.....大特価 ¥15,000

熱転写カラー漢字プリンター (ケーブル付) 送料 ¥1,000

CZ-8PC4 ¥99,800



●48ドット
サーマルヘッド
●B5~B4まで
●ハガキ可能
●カラー対応

オクト大特価 ¥55,800

①CZ-8PC3(24ドット熱転写カラー漢字プリンター)
定価 ¥65,800.....大特価 ¥45,000
②CZ-8PK9(24ピン漢字プリンター80桁)
定価 ¥89,800.....大特価 ¥72,000
③CZ-8PK10(24ピン漢字プリンター136桁)
定価 ¥97,800.....大特価 ¥78,000
④CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価 ¥130,000.....大特価 ¥104,000
⑤CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価 ¥160,000.....大特価 ¥128,000
⑥IQ-735(カラーイメージジェット)
定価 ¥248,000.....大特価 ¥198,000

パソコンラック 推奨 送料 無料

①五段キャスター付 ②四段キャスター付 ③三段キャスター付



5段キャスター付
キーボードが収納できる
から、手でマウス操作が
ラクラクできる
棚板5段のマルチに
活用できるデスク。
ウーン、こいつはデキル!
1325(H)×640(W)
×700(D)
大特価 ¥16,000



4段キャスター付
どんなパソコンにも
フレキシブルに対応し
使い易いデスクです。
1245(H)×614(W)
×600(D)
大特価 ¥12,000



3段キャスター付
場所を選ばない
簡易で便利な
デスクです。
1175(H)×640(W)
×600(D)
大特価 ¥8,800

X68000ソフト大セール実施中※ゲームソフト25%OFF

クラシック	型名	商品名	定価	大特価
●Z's STAFF PRO68K Ver.2.0 (シフト)定価 ¥58,000 オクト大特価 ¥40,000	CZ-21LS	CompuLink PRO-68K	¥39,800	¥28,800
●データベース ●KAMIKAZE (サムシグネット)定価 ¥68,000 オクト大特価 ¥46,000	CZ-21BS	BUSINESS PRO-68K	¥68,000	¥48,000
●クラシック ●C-TRACE68 (キャスト)定価 ¥68,000 オクト大特価 ¥51,000	CZ-21MS	MUSIC PRO-68K	¥18,800	¥13,500
●C言語 ●C & Professional Pack (マイクロウェアジャパン)定価 ¥58,000 オクト大特価 ¥44,000	CZ-21MS	SOUND PRO-68K	¥15,800	¥11,500
●クラシック ●サイクン エキスプレ 定価 ¥78,000 オクト大特価 ¥58,000	CZ-21MS	Sampling PRO-68K	¥17,800	¥12,800
●クラシック ●デジタルクラフト 定価 ¥39,800 オクト大特価 ¥28,000	CZ-21SS	OS-9 X68000	¥29,800	¥21,000
●ワープロ ●ハイパーワード 定価 ¥39,800 CZ-251BS オクト大特価 ¥29,800	CZ-22BS	DATA PRO-68K	¥58,000	¥41,000
	CZ-22HS	New Print Shop PRO-68K	¥19,800	¥14,300
	CZ-22CS	Communication PRO-68K	¥19,800	¥14,300
	CZ-22LS	THE 福袋 V2.0	¥9,900	¥7,500
	CZ-22BS	CARD PRO-68K	¥29,800	¥21,300
	CZ-24BS	システム手帳リフレックス	¥9,800	¥7,500
	CZ-24SS	活用フォーム集	¥9,800	¥7,500
	CZ-24MS	Homan 68K Ver.2.0	¥9,800	¥7,500
	CZ-24BS	MUSIC PRO-68K(MIDI)	¥28,800	¥20,800
	CZ-24BS	Stationery PRO-68K	¥14,800	¥11,500
	CZ-24BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
	EW		¥38,000	¥29,800
	G-68K		¥14,800	¥11,400
	E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限)(送料 ¥1,000)

●CZ-822C(BK)定価 ¥ ? 大特価 ¥ 18,800
●CZ-888C(BK)定価 ¥168,000 大特価 ¥ 69,800
●CZ-601C(BK)定価 ¥319,800 大特価 ¥174,000
●CZ-611C(BK)定価 ¥399,800 大特価 ¥218,000
●CZ-652C(BK)定価 ¥298,000 大特価 ¥188,000
●CZ-662C(BK)定価 ¥408,000 大特価 ¥248,000
●CZ-601D(BK)定価 ¥119,800 大特価 ¥ 68,000
●CZ-601D(GY)定価 ¥119,800 大特価 ¥ 68,000
●CZ-612D(GY)定価 ¥119,800 大特価 ¥ 74,000

店頭ゲームソフト25%OFF! ビジネスソフト 25%より特価中

●尚、送料として1ヶ ¥500、2ヶ ¥700、
3ヶ以上で ¥1,000となります。(税別)

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL: 03-730-6271

お申込みはお電話でお願いいたします。お客様の(住所)・(氏名)・(電話番号)及び(商品名)をお知らせ下さい。●入金確認後、お電話にて商品をお送りいたします。

現金
振込
払い

銀行振込: お近くの銀行より(電信扱い)にて
お振込み下さい。
現金書留: 封筒の中に住所・氏名・商品名を
ご記入の上、当社までお送り下さい。

クレジット

専用お申込用紙をお送り致します。
ので、必要事項をご記入、ご捺印の上
ご返送下さい。手続きは簡単です。

オクト ラック クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込
先

富士銀行 三菱銀行
久ヶ原支店 蒲田支店
①No.1824 ②No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。

※連休のお知らせ=7/31(水)、8/1(水)は連休です。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。

※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

平成2年夏のボーナス一括払いOK!! (8月末)手数料ナシ!!

超低金利クレジットをご利用下さい。

夏ツクモ! ガバーデン 決算セール

掲載商品2万円以上
送料 無料!!

は7/31(火)迄です。

冬のボーナス一括払受付中! くわしくはお問い合わせ下さい。

ツクモ決算! 展示棚ズレ品

SHARP PA-6500
定価 ¥17,800
限定5台
55% OFF

決算特價 ¥9,800

SHARP PA-7000
定価 ¥19,800
限定5台
51% OFF

決算特價 ¥9,800

SHARP CZ-8PC3
定価 ¥65,800 限定5台
24ドット熱転写カラー複写プリンター
80% OFF

決算特價 ¥39,800

SHARP CZ-8PK7
定価 ¥122,000 限定5台
24ピン、80桁
51% OFF

決算特價 ¥59,800

SHARP CZ-8PK8
定価 ¥152,000 限定5台
24ピン、136桁
45% OFF

決算特價 ¥83,800

ツクモ通販受注センターフリーダイヤル

0120(377)999

商品のお問い合わせは各店又は通販部 ☎03(251)9911へ

LET'S MUSIC

Aセット

CM-32L ¥69,000
SX-68M ¥19,800
Musicstudio Mu-1 ¥19,800
合計定価 ¥108,600

ツクモ特價 ¥91,800

(消費税別途 ¥2,754)

クレジット例(税込)月々 ¥5,830(18回払)

★Musicstudio PRO-68K V1.1又は、Music PRO68K(MIDI)のソフトの場合には ¥8,000プラスになります。

Bセット

CM-64 ¥129,000
SX-68M ¥19,800
Musicstudio Mu-1 ¥19,800
合計定価 ¥168,600

ツクモ特價 ¥144,000

(消費税別途 ¥4,320)

クレジット例(税込)月々 ¥7,107(24回払)



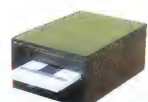
PROII CZ653C 定価 ¥285,000
CZ663C 定価 ¥395,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●知的ニュースタンダードフォーム ●BIOSの改良によりハイスピード処理を実現 ●2Mバイトの大容量メモリを標準装備 ●拡張I/Oポート4スロット標準装備

EXPERTII CZ603C 定価 ¥338,000
CZ613C 定価 ¥448,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●柔軟なフォーム、マンハッタンシェイプ ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

SUPER HD CZ623C 定価 ¥498,000
●次世代のインテリジェンス、SX-WINDOW搭載 ●「チタン」カラーのクリエイティブブラック ●80MBハードディスク搭載 ●世界標準 SCSIインターフェース標準装備 ●BIOSの改良によりハイスピード処理を実現 ●3Mバイトの大容量メモリを標準装備

X68000用ハードディスク

シャープ
光磁気ディスクユニット
CZ-6MO1 予約受付中/
SCSIボード
CZ-6BS1 予約受付中/



アイテック

IT X640 定価 ¥158,000
特價 ¥89,800
IT X680 定価 ¥198,000
特價 ¥118,000



Software tools

GRAPHIC TOOLS

マジックパレット 特價 ¥16,830
Z's STAFF PRO-68K 特價 ¥49,300
サイクロンExpress α68 特價 ¥83,300
デジタルクラフト 特價 ¥33,800

電子手帳ソフト

CYBERNOTE PRO-68K 特價 ¥16,830
Stationery PRO-68K 特價 ¥12,580
※通信ケーブル CE-300L 特價 ¥2,520

電子手帳 & ポケコン

PA-8600 特價 ¥24,800
PA-7500 特價 ¥17,800
PC-E500 特價 ¥24,800



情報 ツール

All in Note



- 「Business Mate」標準装備
- 20MバイトHD搭載
- フリートップサイズ
- 小さいボディに高性能

周辺機器

3.5インチフロッピーディスクドライブ
UE-1F04 定価 ¥49,800
一体型外部バッテリー
UE-1X07 定価 ¥26,000

表計算ソフト

Microsoft EXCEL Ver.2.1
定価 ¥98,000

ワープロソフト

一太郎 AX 定価 ¥68,000
新機AX(UE-6Z10) 定価 ¥49,800

AX286N-H2
定価 ¥398,000

★発売記念特別価格にて提供中!! 詳しくはお電話で!

通信モデム & ソフト

アイワ PV-A24MNP5 ツクモ特價 ¥29,800
(消費税別途 ¥894)
た〜みのる2 ツクモ特價 ¥15,000
(消費税別途 ¥450)

X68000用メモリーボード

一流メーカー

1MB増設用RAMボード ツクモ特價 ¥19,800
(ACE & PROシリーズ内蔵用IMB)
2MB増設用RAMボード 定価 ¥50,000 ツクモ特價 ¥42,500
4MB増設用RAMボード 定価 ¥88,000 ツクモ特價 ¥74,500
※2MBと4MBは全てシリーズ対応拡張スロット用

ツクモグローバルカード

国内・外で活躍!

使って便利、持って安心! ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK! しかも18才以上なら学生でもOK!



お申し込みは(03)251-9898又は各店で



営AM10:15~PM7:00 (休毎週木曜日と8/15)

★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

★商品のご注文は在庫確認の上お願いします。



7号店 荒井



N.C店 堀地

ツクモ7号店 ☎03-253-4199(担当/荒井)

便利で安心な通信販売

通信販売部 ☎03-251-9911

■ニューセンター店 ☎03-251-0987(担当/堀地)
■ツクモ5号店 ☎03-251-0531(担当/川名)
■名古屋1号店 ☎052-263-1655(担当/吉高)
■名古屋2号店 ☎052-251-3399(担当/横山)
■ツクモ札幌 ☎011-241-2299(担当/村井)

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-251-9911へ お電話1本! 配達日の指定もできます。	月々¥3,000以上の均等払いも 頭金なし、夏・冬ボーナス2回 払いも受付中!	〒101-91 東京都千代田区神田 郵便局私書箱135号 九十九電機株通信販売部 oh/△係	事前に☎でお届け先をご連絡下さい。 富士銀行 神田支店(普)No.894047 九十九電機株	くわしくは各店にお問い合わせ下さい。 ケースに合わせてご相談にのらせて頂きます。

表紙ぎゃらりい

1982年5月18日の創刊以来、本誌は誌名を変えても変わらぬ心で誌面作りを続けてきました。応援して下さった読者の皆さん本当にありがとうございます。お

かげさまでOh! Xは通巻100号を数えることになりました。ここにその表紙のすべてをご紹介します。これから本誌をよろしくお願いいたします。

①創刊号



②7月号



③8月号



④9月号



MZ 専門誌としてデビューしたOh!MZ。創刊号は104ページで620円。あまりに高いとの声に次号から480円に値下げしたが……。ちなみに表紙はマッシュパス、オークスターなるヒロインが活躍した。まだXIが誕生する前の時代である。

⑤10月号



⑥11月号



⑦12月号



パソコンテレビXIの登場で誌面に緊張感が。だが、誌名までが変わってしまう事態を予想した人はとれただけであろうか。時はMZ-700の全盛期。一時は読者の4割を超えることもあり、本誌は飛躍的な部数アップを記録した。

⑧1月号



⑨2月号



⑩3月号



⑪4月号



⑫5月号



⑬6月号



⑭7月号



⑮8月号



⑯9月号



⑰10月号



⑱11月号



1912月号



4月号からあのシド・ミードが表紙を飾る。増ページと共に内容も充実し、ほぼ現在のスタイルを確立。そして11月号には新製品X1 turboの歴史に残る大特集が。MZユーザーの目がこれ以来反感から羨望へと変化したという。

201月号



212月号



223月号



234月号



245月号



256月号



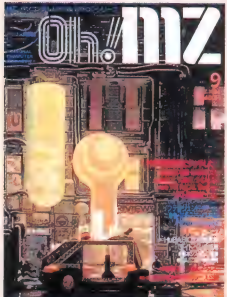
267月号



278月号



289月号



2910月号



3011月号



3112月号

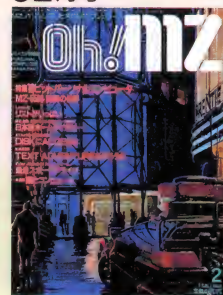


感動のX1 turbo特集

321月号



332月号



343月号



364月号



375月号



386月号



397月号



408月号



419月号



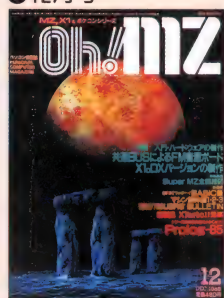
42 10月号



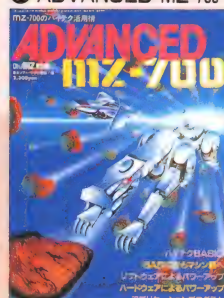
43 11月号



44 12月号



35 ADVANCED MZ-700



本誌唯一の別冊。発売
が遅れてMZ-700のユー
ザーをやきもきさせた。

45 1月号



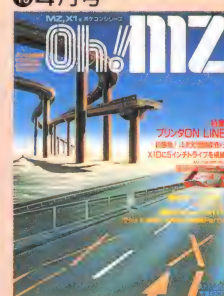
46 2月号



47 3月号



48 4月号



読者参加を強く呼び掛ける
特別企画「GAME OF THE
YEAR」「言わせてくれなくち
ゃだワ」を開催。このまま
では世のパソコンがすべて
実務マン一色になるとい
う不安のなか、ついに夢の
マシンX68000が衝撃のデビ
ューを遂げたのだった。

49 5月号



50 6月号



51 7月号



52 8月号



53 9月号



54 10月号



55 11月号



56 12月号



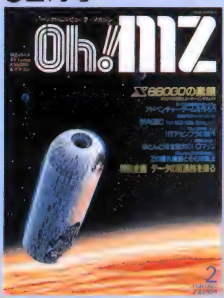
57 1月号



MZ-286iを機にMZグルー
プがビジネスコンピュータへ
の路線転換、パーソナルユ
ースはXfamilyに絞られる。
そのため本誌は12月号でOh!
MZ→Oh!Xと改題した。なお、
1月号から翌年3月号まで
の表紙イラストは永沢しげ
る氏が担当。

X68000が初登場！

58 2月号



59 3月号



60 4月号



61 5月号



62 6月号



63 7月号



64 8月号



65 9月号



66 10月号



67 11月号



68 12月号



X68000ユーザーが増えるなか、本誌では創刊6周年企画として8TRON計画を発表、昭和70年代を目指した究極の8ビットパソコンの姿を考えた。結局70年は来なかったが……。また、4月号からは画家の松葉口忠夫氏に表紙絵を依頼。

これが初のOh!X

69 1月号



70 2月号



71 3月号



72 4月号



73 5月号



74 6月号



きわどい内容が満載

75 7月号



76 8月号



77 9月号



78 10月号



79 11月号



80 12月号



81 1月号



82 2月号



83 3月号



84 4月号



ぶっとんだ
ゲーム特集
が衝撃的!

名実共にパーソナルマシンの一大勢力に成長したX68000。読者の割合も半数に達し、誌面もX68000を中心にゲーム、グラフィック、サウンド関係の華々しい記事が目立つようになる。4月号からの表紙はもとのりゆき氏にお願いした。

85 5月号



86 6月号



87 7月号



88 8月号



89 9月号



90 10月号



91 11月号



92 12月号



93 1月号



94 2月号



95 3月号



96 4月号



4月号から表紙デザインを一新。須藤牧人、塚田哲也両氏のCGが交互に本誌を飾るようになった。さて、'90年代のOh!Xは、など能書きを垂れている暇はない。時代はリアルタイムに動いている。Oh!Xはどこへ行くのか？

98 6月号

ついにディスクが付録



97 5月号



1月号付録のX68000ゲームソフトウェアカタログ

お祝いの言葉

へーえ、100号？ そうか、まだ100冊しか出てなかったのか、もっといっていいと思ってた。まあ、100冊たって、数字なんてどーでもいいことさ。さるお方の結婚式ももうすんだし。過去も未来も似たようなもの。大事ななのはその100冊に散らばる過去の名作たちだ。逆立ちしてもOh!X(Oh!MZ)でしか読めない、機種の壁を越えた名作・奇作・珍作の嵐。これが財産である。Oh!X傑作集を出したいくらいだ。

いま、その個性も矢面に立たされている。浸透は常に拡散を伴うからだ。いくつものベクトルを内包した新しいスタイルも必要とされるだろう。しかし、知識より知恵、実用より心、完成されたプログラムよりマシンポテンシャルの開拓精神の基本は変わらない。X68000はまだ深いポテンシャルを秘めている。のんびりしている暇はない。そして粋なパソコン誌として、多様化する読者と共に、Oh!Xは100万部を目指すのである。

からころも きつつなれにし つましあれば
はるばるきぬる たびをしぞおもふ
てなもんだ。めでたいな。(荻窪圭)

100 8月号



99 7月号



6月号付録の創刊8周年記念PRO-68K

SOFTWARE INFORMATION

今月は夏休みに向けてか、ひさびさに大量の新作の情報が入ってきました。てなわけで、今回は4ページでお届けすることになります。しかし、毎月コンスタントにこのくらい発表されればありがたいのに……。



ギャラガ'88

2, 3年前だけどゲーセンで流行ったこのゲーム、いよいよX68000にも登場だ。ゲーセン版の移植のみならず、X68000オリジナルの面もあるぞ。



話題のソフトウェア

いや～、先月は梅雨だなんて書いてしまったもんだから、皆さんからのお叱りのハガキの多かったこと。まあ6月18日を予想して書いているんだから、そーゆーこともたまにはあるわな。許せ許せ、ハハハ。というわけで、今月こそ梅雨です。じつにうっとうしいですね～（え？ フォローになってないって？ でも、梅雨明けって7月22日って気象庁が言ってるからいいじゃない）。そういや、もうじき夏休みですねえ。クーラーの効いた涼しい部屋でアイステイでも飲みながら、ゆったりとゲームに浸る。う～ん、極楽極楽（とか言ってすっかり違う方向へ話を持っていくヤツ）。悪いことは全部忘れて、夏休みの前半は遊びまくりましょ。宿題そのほかで青くなるのは、来月号が出てからでも十分なんだから……（ホントか、おい）。

さて、夏休みを目前に控えて、ゲームのほうもバタバタと活気を増してきました。

なんともうれしいぢやあ～りませんか。うれしさ爆発、ページも倍。これを書く側としては、ほんっとに喜んでいいやら悲しんでいいやら……。ま、そんなこと言ってもしょうがないんで、順を追って紹介していくことにしましょう。

まずはこのギャラガ'88。電波新聞社よりすでに発売されているので、もうクリアしちゃった人もいるんじゃないかな。このゲーム、3年ほど前にゲーセンで流行ったナムコのシューティングなんだけど、たった3年前なのに第一印象で“懐かしい！”と感じてしまいました。もっとも私の場合はこのゲームの元祖、ギャラクシアン（死語だよなあ）を中学生の分際ながら（あ、年がバレる）ゲーセンで遊んでたから、そのとき印象が強いからかもね。で、肝心の出来ですが、これがなかなか。プーッとふくれるハエさんや、かわいいボーナスステージのギャラクティックダンシングもゲーセン版同様のいい味出しています。さすがに先に移植されていたPCエンジンよりは、グラフィックもきれいですし。これはゲーム自体は、そう難易度の高いシューティン

がんばで、ぐただだ！

1	ポピュラス	(前回順位)	1
2	グラナダ		4↑
3	ワンダラーズ・フロム・イース		3
4	ダンジョンマスター		2↓
5	天下統一		一初
6	スーパーハンガオン		一↑
7	ジェノサイド		10↑
8	三国志II		5↓
9	サーク		6↓
10	ソーサリアン		7↓

疲れたー。いつもはサンプリング抽出をして

るのに、今月は28日までのハガキを全部カウントすることになってしまいました。手伝ってくれたみんな、ありがとね。

さて、100号記念(かどうか知らんが)の完全集計版TOP10。ランクアップ・ダウンもつけてみたけどどうでしょう。

おやおや。そろそろみんな解き終わったと思ったらダンジョンマスターは4位まで落ちてしまったぞ。みんな結構ドライだな。代わって2位の座を手に入れたのは、グラナダ。これはウルフ・チーム最高順位！ イースファンのみなさん、もう少しだったのに、残念でしたね。

そして、5位初登場天下統一。このゲームの



バズニック

ぐでもなかったの、ゲーセン版のほかにも、X68000用にオリジナルステージも用意されています。こちらもぜひプレイしてみたいですね。

さて発売中といえば、ブロードバンド日本の**バズニック**。こちらもゲーセン版（タイトー）からの移植です。ゲーセンではじっくり考えているヒマがなかったので、かなりお金を注ぎ込んだ人もいたことでしょう。同じマークのブロックを隣接させて消していくパズルゲームなのですが、ブロックは重力の関係で上にあげられないし、でもってタイミングが命の面もたくさんあるし、一筋縄ではいかず悩むわけなんです、これが。家でじっくり楽しめるようになれば、クリアも夢じゃなくなるかな。でもムリかな、私バカだから。

でもって、同じパズルゲームであるコナミの**クオース**ももう発売されていますね。こちらもゲーセン版からの移植もの。ゲームボーイなどでも発売されているし、けっこうやり込んでいる人もゴロゴロいるのでは？ このゲームはシューティングの要素も含まれているので、ちょっとだけ反射神経が必要かもしれないけど……。

ん？ こうやって書いていくと、なんかゲーセン版からの移植ものばかりだわねー。ま、いっか。ついでだから、このまま続けて移植ものを一気に書いてっちゃおうと。

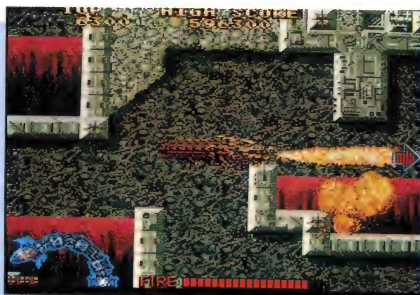
じゃ、次、**サイバリオン**。このゲームはドラゴンを操って、矢印の指し示す方向へ

評判は……あれ、ハガキはAFTER REVIEWに行っちゃったの？ じゃあすみません、そっちを見てちょうだい。

その下に謎のカムバック、スーパーハングオン。確かに長く遊べるが、なぜ今になって……。さらに7位ジェノサイドのランクアップも謎だ。もうすぐラグーンも発売されるというのに……。そういや、みんなCDはもう買ったかな。

あやや、三国志IIもソーサリアン（まだいる！）もランクダウンか。先月威張ったのが反感を買ったかな？ こりやおいらは静かにしてたほうが良さそう。……（それじゃ、また来月）。

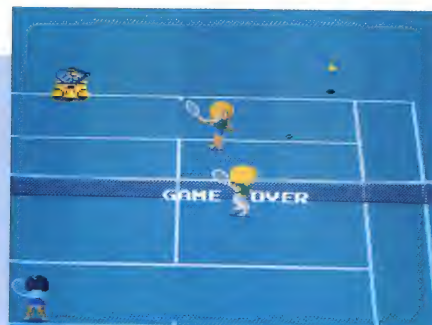
（浦）



サイバリオン

進んでいくタイトーのアクションゲームなんだけど、ゲーセン版はスティックじゃなく、トラックボールで遊んでるところがミソだったよね。今回はジョイスティックでもできるようにになっているけど、通ならやっぱトラックボールで遊んでほしいな。ジョイスティックに慣れているからこそ、トラックボールで遊ぶっていう感覚は新しくっていかもしないし。8月中旬にシャープから発売される予定。いま頑張ってるSPSさんが移植しているので、楽しみにしてて。

でもって、同じくSPSさんの移植によるナムコの**ワールドコート**の登場です。このゲームってば、地味なスポーツゲームと思いきや、結構ハマりやすいゲームだったりするわけ。その当時は友達同士で遊んでいる高校生や予備校生をよく見掛けました。そうこうする間に、PCエンジンにも移植されちゃったりなんかしました。さすがに今回はクエストモードはないみたいだけどね。スマッシュやサーブがうまく決まるようになると、もうまさにテニスの選手になった気分です。そういや、わざと女の子の選手を転ばせてパンチラを楽しん



ワールドコート

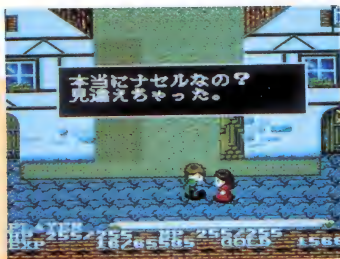
でいたふとときものもいたっけかなー。まあ、それはおいといて、このゲームは7月20日に発売される予定ですのでお楽しみに。

さてお次は、じゃーん、**イメージファイト**なんですねー。このゲームはかなりムズかったんで、わりとマニア受けしていたシューティングです。アイレムさんのゲームはあのR-TYPE以来だから、このイメージファイトの登場を待ち望んでいたユーザーも結構いるはず。その夢がやっと実現しました。このゲーム、ポッドと呼ばれるアイテムを、いかにうまく使いこなすかがカギとも言えるでしょう。これをうまく扱えないと、かなり苦しい。はじめてやると全9面クリアどころか、5ステージクリア後にある補習ステージにたどりつくのにもてこずったりするんですよ、これが。で、移植の出来はというと、画面写真を見てのとおりに。なかなかよさそうでしょ？ コンテニューもあるらしいから、ゲーセン版では見ることができなかったエンディングも見られるかもしれないぞ。年内発売の予定だから、詳しい情報はもうちょっとだけ待っていてね。

イメージファイト

これまたゲーセンで人気だった超ムズいシューティングゲーム。なかにはゲーセンで血を流した人もいるとかいえないとか……。





ラグーン

ジェノサイドで人気のソフトハウス、ズームの期待の第2作。今度はアクションRPGだぞ。2頭身のキャラクターがなんとも可愛い。期待度大のゲームだ。



実戦ビリヤード

まあ、ゲーセンからの移植情報はこんなもんな。もうちょっとすると、またいくつか出てくるみたいだけど、それはそれでまたあとのお楽しみということで、ね。

じゃあ、今度はゲーセンものではないやつをガシガシ紹介していくことにしましょうか。

まずは、皆さんお待ちかねのズームのラグーンからいきましょう。ジェノサイドで一躍人気者となったズーム。その第2弾といえば、アクションゲームファンでなくとも気になるところ。開発状況はわりとよいようで、発売に向けて着々と進行している様子です。今回は、最終段階に入ったともいえる現時点での画面写真をお届けしましょう。ジェノサイドであれだけ頑張ってくれたズームが、アクションRPGという新境地でどういった展開を見せてくれるか、楽しみにしたいですね。

さて、バトルチェスでX68000に参入したバック・イン・ビデオからは、実戦ビリヤードが発売中。このゲームは、その名の通りビリヤードゲームで、ナインボールやローテーション、はたまた4つ玉(知ってるかな?)までプレイできちゃいます。プールバーなるものが乱立したビリヤードズームはもう過ぎてしまいましたが、本来

ビリヤードというものはじっくり玉筋を読んで楽しむものだし、家でゆっくりビール片手にパソコンに向かって楽しむのもいいんじゃないでしょうか。

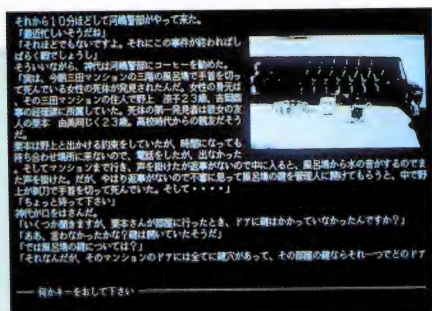
じっくり楽しむといえばやっぱりMisty4でしょうか。一連のMistyシリーズの第4弾です。前作からしばらく間が空きましたが、やっぱりデータウエストさん、頑張ってくれました。今回はユーザーからのシナリオ5つを中心に構成されてます。暑い夏に、ちょっとサスペンスタッチの推理ゲームを静かに楽しむ、なんて大人っぽいじゃない。ところでデータウエストといえば、ピンとくるのが第4のユニットシリーズ。ブロンウィンファンの皆さん、ご安心を。シリーズ第5弾D-Againも着々と進行している様子。今月はまだ画面写真をお届けできないけど、もうちょっとしたら詳しいことをお伝えできそう。待っててね。

でもってT&Eからはルーンワース〜黒衣の貴公子〜が発売、ドラマチックな展開で進んでいくアクションRPGです。なぜドラマチックかというと、このゲームはプレイヤーの行動によって、たどるストーリーが変わっていくからなんです。いわば、あなたがストーリーを作り上げていくゲームなのです。うん、これは奥が深いぞ。

またT&Eでは次回作幻獣鬼を開発中。これはサンプル版をプレイしたところによると、敵の攻撃が、というか敵の放つ弾が雨アアレのごとく飛んでくるので、なかなかタイヘン。やりがいがありそうです。そのほか、あのゴルフゲーム遙かなるオーガスタも出す予定だそうだし、今後のT&Eの動向には目が離せない!?

さてさて、数々のラインアップを控えているザイン・ソフトでは、ただいまREINFORCERとパルーサの復讐をしゃかりきになって開発中。REINFORCERのほうは、トップビュータイプの8方向スクロールという、サイバーパンクアクションゲームだそう。こちらは先月号でも紹介しましたが、さらに開発が進んだものが手に入ったので紹介しちゃいましょう。発売は9月上旬の予定。一方のパルーサの復讐のほうは、剣と魔法で攻撃するファンタジーアクションゲーム。サイドビュータイプで、8方向多重スクロールするというシロモノ。こちらは7月発売を目指して、目下頑張っている中。お楽しみに。

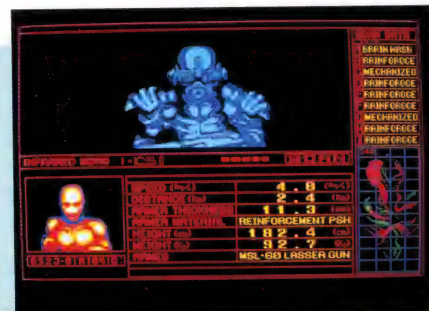
あつ、とついうっかり忘れそうになっちゃった、いまや読者の人気ナンバー1に輝いたポピュラス。そのポピュラスの追加シナリオが発売になったことは、きつともう皆さん周知の事実でしょう。今号のREVIEWでも紹介していますしね。まあ、それはおいといて、なんとそのポピュラスを発売したイマジニアから、シムシティが移植、発売されることが正式に決定しました。



Misty4



幻獣鬼



REINFORCER



闇の血族

サコムのノベルウェアシリーズ。推理探偵もので主役はうら若き乙女。リアルな感じのグラフィックが雰囲気を出しているよね。

わーい、パチパチパチ。このシムシティー、都市開発を題材にしたリアルタイムシミュレーションで、14個のアイコンを駆使して町を発展させることが目的。鉄道を敷いたり工場を建てたりとなんとも忙しい。まあ、詳しいことはまた来月にでも紹介させていただきますのであしからず。へへっ、出し惜しみしちゃってごめんね。また、イマジニアではポピュラスの原作者であるピーター・モリニュー氏の来日を記念して、ポピュラス大会を企画しています。我こそは、と思うポピュラスマニアの方、んあ？ と思ったらプロミストランドのREVIEWの左下を見て、応募してください。よろしくね。

さて、移植といえばスタークラフトのトンネルズ&トロールズ。こちらもすでに発売になりましたね。もともとテーブルトークRPGということで、そのあたりが好きな方には熱狂的な支持を受けているゲームですが、ようやくX68000にも登場。ほっとした方もいることでしょう。このゲームは、背景となる舞台設定がしっかりしているので、はじめてRPGをやる人でも親しみやすいかな。それにオマケとしてオリジナルオーナーズカードや、ドラゴン大陸のポスターなど、RPG必携3点セット(!)なるものが付いてくるなど、ニクい心配りがうれしいじゃありませんか。毎日コツコツとたゆまぬ努力をしても苦にならない方は、ぜひプレイしてみてもは？

あちらものの移植じゃあないけれど、こちらも移植もの。PC-9801からの移植だけれど、システムソフトから遊撃王IIがでるそうです。PC-9801版ではサイバースティックが使えるってんでびっくらこいたわけですが、当然のことながらこのX68000版でもサイバースティックが使えます。フライトシミュレータゲームなので、サイバースティックを使えば、パイロット気分が楽しめそう。画面写真もお届けできなかったし、発売はまだ未定だけれど、出来はか

なりよさそうですよ。期待度大です。

さてと、そいじゃシステムサコムだ。ジェミニウイングの開発も佳境に入ったカンジなのだけれど、その一方であのノベルウェアシリーズである闇の血族の開発も、しっかり進行している様子。今回は女の子が主役のアドベンチャーとあってか、サコムとしても主人公のグラフィックにはリキを入れているよう。届いたばかりのグラフィックの数々を紹介しましょう。この闇の血族は、7月か8月には発売されるそうなので、ノベルウェアファンは見逃せませんね。

さて、最後を飾るのはM.N.M.Softwareです。今回紹介するのはThriceとPipyan。ThriceはColumnsタイプのパズルゲームで、縦、横、ナナメに同じキャラクタを3つ以上揃えて消していき、得点を競うというもの。なんと300位までネームエントリーができるそう。ふえ〜。でもって、このタイプはずっと画面を見ているだけでは疲れてきちゃうこともあるので、それをな

くすためにもある点数をクリアするごとに、背景がいろいろと変わっていくので、飽きずにプレイできます。8〜9月に発売されるそうなので、Columnsにはまった人はぜひプレイしてみてください。そしてPipyanは、倉庫番のように男の子のキャラクターを操作して、ブロックをうまく組み立てていくといったゲームです。さながら工事現場のようなステージ上で、あたふたと動き回る男の子、失敗するとペコペコと頭をさげたりなんかして、とってもキュート。こちらは、7月中旬にタケルより発売される予定とのこと。ひょっとしたらこの本が出る頃には発売されてるかもね。

てな感じで今月もそろそろネタ切れです。こうやってずらっと書いたあとで見てみると、おや、X1がひとつもない？ んなバカな！ でもほんと、そうみたい……。なんかとっても悲しいなあ。ああ、X1ユーザーの怒りの声が聞こえてきそう。ではまた来月。



トンネルズ&トロールズ



バルーサの復讐



Thrice



Pipyan

THE SOFTOUCH

●大航海時代



ロマンたっぷり 大海原で帆船の冒険

Urakawa Hiroyuki
浦川 博之

「維新の嵐」に続く光栄のRÉKOEITION GAME第2弾は、中世の帆船の旅をシミュレーションゲームにした「大航海時代」です。貿易、艦隊との対決、数々の使いっぱ(?)を繰り返し、成り上がるのが目的だっ!



X1turbo用 5"2D版 4枚組 9,800円(税別)
光栄 ☎044(61)6861

ども。親父が船乗りだった浦川です。おかげで家はいろんなオブジェでいっぱい。おさるさんの置物とか巨大な素焼きの風鈴とか、ダチョウの卵とか。こう節操なく並ぶと海のロマンもなにもあったもんじゃない。で、その因縁か、私が光栄の「海のロマンゲーム」、大航海時代のレビューをやることになりました。これは1500年代初頭を舞台とした海洋シミュレーションです。ポルトガル、イスパニア、イスラムによる貿易の主導権争いの真っ只中の頃ですね。プレイヤーは有象無象の商船長の中のひとりとなり、地中海に始まって、アフリカ喜望峰、アラビア、インド、はてはジパングまで航路を開拓し、貿易を行います。

貿易のほかにもうひとつ、貴族の爵位を得るというフィーチャーがあります。主人公の先祖が航海の失敗から爵位を剥奪されたという設定になっていて、お家の復興がプレイヤーの悲願なのです。オーイェー(面白度1)。ライバル国の艦隊をやっつけたり、勅命を遂行したりして国王に認めてもらい、最高爵位まで昇りつめるべくこれまた世界を駆け巡るわけですね。

ややこしそうに聞こえるかもしれませんが、「貿易する“スタークルーザー”」といえはわかるかな(もしくは光栄版“WARNING”か?)。

地中海の隣人

私はタバスコ＝ガマ。ちょっと辛口のいい男。自分ではちょっとだけ銀英伝のラインハルトに似てると思っている。親父が遭難して行方不明になったので、家の再興のために大海原に出て一旗上げることにした。といっても、手元にあるのは親父の残した小さい商船だけ。最初はヨーロッパ周辺で経験を養い、財力をつけねばならない。幸い、頼りになる昔の父の部下ロッコがいる。ひとりでもロッコとはこれいかに? ロッコ「ぼっちゃん、禅問答してないでこれからどうするか決めてくださいよ」

じゃあ酒場に行こう。情勢も知らずに積み荷を仕入れちゃ失敗は目に見えてる。カランコローン。

Yo「あら、いらっしやい。」

ようこちゃん、ここのみんなにwellsスーパーマラソンね。

ロッコ「おや、誰か来やすぜ」

男「あんた、リスボンで何か仕入れるんだったら、砂糖を買うといいぜ」

かくして1502年2月、タバスコ一行と砂糖をどっさり載せたラテン船「難破1号」は大西洋へ漕ぎ出した。……誰だ、こんな不吉な名前つけたやつあ。

航海中の画面は下の写真のとおり。1画面が緯度・経度ともに約5度の広さだ。この左側の矢印はなんだろう。

ロッコ「上は針路。真ん中は風力計でさあ。左上の数字が風力で、その下は潮流計」

いまは逆風だな。三角帆だから逆風でもわりと速いんだよな。速い速い……(ゆるゆるゆる), 速い……。おい、遅いぞ。なんだこの遅さは。おまけに夜が明けるたびにディスクはガーガー鳴るし。

ロッコ「この辺りは外洋と違って風がおとなしいですからね。それに海を航行してるのはわしらだけじゃねえんから、処理速度もちったあ遅くなりまаса」

ぶーぶーいいながら、3日でイスパニアの首都、セビリアに到着。幸い、砂糖は約2倍の値段で売れた。元が安いからあまり大きな儲けにはならないが、楽な航海だったからこんなもんだらう。しかし、どこの港も人の顔が全部一緒だな。旅情ってもんがない。酒場の娘の顔は違うんだけど。

ロッコ「なにぶつぶついつてんです。次はどこへ向かいやすか?」

神聖ローマ帝国のピサで美術品が安く買えるようだから行ってみよう。

再びゆるゆると地中海を進む。このゲーム、舵を切るときはメニューを開かなくてはならない。そのたびにディスクアクセスするので、地中海のような入り組んだところを航行するのはなかなか骨が折れる。

十数日の航海を経て、ピサに到着。すいませーん、美術品くださーい。

交易所の親父「美術品は金貨310枚だよ。いくつ買うかね?」

買えるだけ全部。ところで、この美術品って中身はなんなの?

親父「見てみるかい(ごそごそ)。ほら、



航海中の画面はこんな感じ、どんぶらこっと

名物“ピサの斜塔ぶんちん”。いまなら大小の鉄球もつけちゃう」

ガリレオの実験は100年後なんですが……。

ザ・グレートスト・ミッション

半年近く地中海を駆け巡ったおかげでめでたく2隻目の船を購入できた。名前はもちろん“難破2号”。途中酒場で知り合ったオスワルドという男に船長をまかせる。

地中海の主な貿易ルートは次のとおりだ。

- ・リスボン（砂糖）→セビリア
- ・アントワープ（陶磁器）←→ロンドン（羊毛）
- ・ピサ（美術品）←→マジョルカ（穀物）

もともと、港ごとに物価は違うし、ほかの艦隊の取引によっても相場は変動するので、絶対これというパターンはない。それから「イスタンブールの美術品はいいぞお、儲かるぞお」とさんざん吹きこまれたが、ポルトガルとイスラムの仲が悪いので立ち寄っても追い返されてしまった。王様、なんとかしてよ。トホホ。

さて、そんなある日。立ち寄った酒場で見知らぬ男に呼び止められた。

男「よう、あんた。タバスコさんだろ。マジョルカであんたを捜してる奴がいたな」
ロッコ「なんででしょうね、ぼっちゃん？」
デ、デートの申し込みかな？（ずで）

耳を引っばって連れていかれたマジョルカ港では交易所の親父が待っていた。

親父「わざわざどうも。あなたに頼みたいことがあって捜していたんです。実は陶磁器で儲けようと思うんですが、35ほど仕入れてきてもらいたいんです。金貨4620枚で仕入れてきてもらえますか？」

わざわざ呼びつけて使えばかよー。

ロッコ「そういうことってちゃいけやせん。かなりワリのいい仕事なんすから。それに交易所御用達になれば王様のお目に止まる日も近いですよ」

ぶーぶーいいながら申し出を受け、ヴェネチアで陶磁器を仕入れてくる。さっさと引き渡し、その報酬で飲んでいと……。男「おい、タバスコさんだろ。リスボンで

王様がお呼びだったという話だぜ」

ロッコ「やりやしたね、ぼっちゃん！すぐに駆けつけやしょう」

もちろんだあ。この家名復興のチャンス逃がしてたまるか。リスボンに急行だ！

ゆるゆるゆる。リスボンを目指して帆船はのんきに進む。リスボンに着くや否や、一目散に城へ駆けこんだ。

役人「謁見の申し込みか？しばらく待たれよ。……陛下がお会いになるそうです」

荘厳な謁見の間に通される。国王が現れた。面を上げる。緊張の一瞬。

ポルトガル国王「おお、そなたがタバスコか。お前を呼んだのはほかでもない。実は羊毛が38必要なのじゃが、そなたに……おいおい、どうしたのじゃ？」

タバスコ、南へ

勅命の使いっぱを完遂した私は子爵の称号を賜った。あれからイギリス、北欧まで足をのばし、貿易網はイスラムを除いたヨーロッパを網羅している。新たに中型の船を購入して旗艦とし、ポチョムキン号と名をつけた。

さて、ロッコ、新しい船も手に入ったし、ここでひとつアフリカに行ってみようと思うのだが。あそこじゃ金が手に入るという話じゃないか。

ロッコ「うーん、ちょっと装備が弱い気もしやすいが、いつまでもヨーロッパでもないですしねえ」

よし、決まりだ。食料と水を満載し、ひたすら南を目指す。セビリアから2,3日ほど行くと海の色も変わり、アフリカに入ることがわかった。ちなみにBGMも変わる。おお、風が強くなってきたぞ。わあ、強い強い。風力8だ。暴風だぞ、こりゃあ。

ロッコ「これが外洋の風でさあ。これに乗って一気に南下しやすぜ」

ててててて。信じられないペースで船は進んでいく。うわあ、揺れる揺れる。きぼぢわるい、げろげろ。ちょっとアフリカは

〈ちょっとひとこと〉

貿易が題材のゲームというのは、どうしても単調になりがちです。イベントなどを設定してうまく防いでいますが、操作性の問題が目につきやすい序盤では「ずーっとこんなことが続くのか」と目まいを覚えてしまうこともあります。

地中海を出るようになれば、自分で航路を開く楽しみもあって、自分の好きなように遊ぶことが可能になります。規制が緩く、自分の好きなように遊べるのが身上です。なんだかんだいって結局ハマってしまうのが光荣のすごいと



酒場は大事な情報源、そのほかにもいろいろ……



王女クリスとの密会、たまにはこういうのもね

早すぎたかなあという思いが頭をよぎる。

ロッコ「ぼっちゃん、港が見えやす」

え？もう着いたの？まだ1週間そこそこののに。しかし、交易所には金がいっぱい！有り金はたいて全部買い込む。はっはっは。帰れば大金持ちだぞ、ロッコ。

てててて。帰日も快調。見事アフリカ金貿易航路が開けたかと思われたが……。

「提督。嵐だ！」、ざぞーっ。もりもりと海が盛り上がり、船はひっかきまわされた。「舵がききやせんぜ！」。西を向きながら、船は東へ押し流される。もうムチャクチャ。「難破1号の姿が見えやせん！」

海は一昼夜荒れ狂い、さらに難破2号までが行方不明になった。やはり名前が悪かったか。旗艦ポチョムキンも食料の半分と3分の2近い乗組員を失った。安易に外洋に来るんじゃないかった……。と、放心状態でさまよっていたのも東の間。

「提督。嵐だ！」

この船の末路が私の脳裏をよぎった。

ころ。

最後にBGMのことで、音楽性がないとはいませんが、「3パートしか使わないBGMを聞かせてCDを売り込むのはちょっと無理があるんでないの」ということは指摘しておきましょう。

冒険心刺激度	10
マニュアル親切度	9
グラフィック	8
操作性	6
BGM	4
熱中度	8



戦闘画面はやっぱりヘックス

●ウルティマV



天下無敵の シリーズ第5弾

Ogikubo Kei

荻窪 圭

その面白さがわかる人にはすごく面白い。そういう一風変わった、しかも奥深い魅力をもつウルティマシリーズの5作目がいよいよ登場。さらなるリアルさと難解な謎であなたの頭を悩ませる？



X68000用 5"2HD版2枚組 9,800円(税別)
ポニーキャニオン ☎03(221)3161

ああ、ダンジョンマスターって、なんて楽なゲームだったんだろう。メモを取る必要はほとんどなかったし、地下6階までは下を目指して進んでいけばよかった。

ウルティマはそんなわけにはいかない。右も左もわからない大陸の真ん中に放り出され、行くも地獄行かぬも地獄、森の木陰でドンジャラホイ、なのである。世界の合言葉は森ってなものだ。

どーして怠慢で出不精で睡魔に魅入られた私がウルティマVなどという超大河、スーパー大河なゲームをすることになったのかというと、ウルティマIV経験者がほかにいなかったからである。経験者っていうだけで終わらせたわけではなく、しかも3年前、友達の部屋のXlturboIIで遊んだものだったりするので、当然育てたキャラクターは持ってこれないし、当時集めた膨大なメモは引越しの際にみんな捨てちゃったしの後悔先に立たず、あとの血祭り村祭り、かんなん汝を玉にするってな状況。人生、蜜のように甘くタバスコのように辛し。

懐かしい風景、旧知の友

イオロ、シャミノ。記憶の底にこびりついた青春の残滓から消え去る寸前のデータベースにこびりついてた懐かしい名前。こんなことまで覚えているなんて。いや、覚えているというより思い出すことができるといったほうが正しい。あくまでも画面にその名が記されたとき、懐かしさを感じるだけだ。役に立たない記憶。

主人公はアバター。アバターというのはAVATAR、アバターとかアヴァターラなどともいう。「化身」とか「権化」という意味である。化身といえばレインボーマン。レインボーマンは月の化身、火の化身など7種類の化身になれた。つまり、アバターだったわけである。レインボーマンといえば「インドの山奥で修行」。このインドがポイントでアバターというのとはもともとインドの言葉だったのだ。インドにおいてヒンズー教のヴィシュヌ神は人々の前にさまざまな動物や人の姿を借りて現れると考えられ、それを化身(民衆を救おうとして神が姿を変えて現れること、あるいはその姿)、つまりアヴァターラと呼ぶのだ。

で、ウルティマVの主人公はウルティマIVで8つの徳をすべて極め、アバターとなった者なのである。私はなった覚えがないがなったらしいのである。なった覚えがある人(つまりウルティマIVからキャラクター

を移した人)は、それなりのレベルから始められるが、私のようにアバターになった覚えのない人はアバターのくせにレベル2という苦難の始まりとなる。弱い弱い。

舞台はウルティマIVと同じ広大な大陸だ。しかし、前作でとったメモがない。最初からやりなおし。それでも歩いているとだんだんと思い出してくる。ここに村があった、この辺にムーンゲートが出るはずだと。

自由の持つ厳しさ

ウルティマがほかのRPGと異なる点はゲームを進めるためのガイドがまったくないことである。イースを代表とする日本式RPGはスゴロク型であった。ダンジョンタイプのRPGも、その存在自体にダンジョンを深いところへ向かって降りていくという不文律のガイドがあった。しかし、ウルティマは恐ろしい。前向きRPGではなく、はなから、大陸の真ん中で右往左往、どこから手をつけてどこへ向かうのかも自由なのだ。かなりレベルが上がった後半にならなければ行けないような場所でも、然るべき情報と金を出して買えるアイテム(船など)があれば行けてしまうのだ(ちなみに、キーバッファはたまらないぞ)。

つまり、ドラクエやイースやらのスゴロク型RPGが管理された、安全だけど自由のない日本であれば、ウルティマは自由だけど危険ですべて自分の集めた情報を基に自分の判断で動かねばならないアメリカなのだ! ほほほほほう。あなたはどっちが好きですか。自由社会? でも、自由の旗のもとで自由に生きていくためのプレッシャーは相当なものである。

たとえば、ウルティマではお城のオークの樽に隠されたアイテムを盗むことも、寝ている衛兵を殺すことも簡単だ。本当に簡単だ。しかし、その結果がどうなろうと自分の責任である。特に、ウルティマVは平和で善良な人々ばかりであったIVと違って



やったー、ついに亡霊登場でレベルアップだ

邪悪なブラックソーンの支配下にあるのだ。その中でアバターとしての行動をやり通さねばならない。不当な要求に答えて「持っている金の半分の衛兵に支払う」のも、信念を貫いて「牢獄にぶちこまれる」のも自由だ。

うーん。このゲームは「うんちやうんちやらの自由」を要求するガキの精神に「自由の持つ厳しさ」を叩き込む教育ゲームだったのか。私はもちろん、血反吐を吐きながらも、管理された健全な社会よりアナキーで自由な社会のほうを選ぶ。日本という平和で安全な社会が好きな人はガイドに沿って大陸を旅するドラクエでもやってください。

複雑怪奇な社会

ウルティマVには表の世界と裏の世界がある。表の世界がブラックソーンに支配された圧政の社会であり、裏の世界はロード・ブリティッシュに忠実な人々が集まった、レジスタンスである。レジスタンス、そんなものまであるのだ。アバターである主人公とウルティマIVとともに戦った仲間たち。もちろん、レジスタンスとともに行方不明になったロード・ブリティッシュを捜し、この世界に平和と徳を取り戻すのだ。それが目的だ。それにはアバターはアバターらしく行動せねばならない。ものを盗むな、罪のない人は殺すな、邪悪な者に対しては勇敢であれ。

何が自由だ！ 道徳的であらねばいけないなんて！ 規範だらけではないか。しかも目の前にはおいしい餌がぶら下がっているというのに、道徳的であるために自らを律せねばならないのだ。目の前の快樂に弱い荻窪圭はどーしたらいいのだ。

昼と夜

話はがらっと変わる。ウルティマVのうりのひとつに、時間がある。街の住人は朝になると起き、働き、昼になると食事をし、夜になると寝る。だから、買い物をすると思ったら店が開いている時間に行かないと売ってくれない。夜になると門を閉められて入れない街もある。みな働き者で規則正しい生活を送っているのだ。なんと、夜になると会合を開いているレジスタンスの農民もいる。門番の衛兵もちゃんと食事どきや交代時間には入れ替わる。ベッドももちろん住民の数だけある。私は宿屋のない街では他人の家の他人のベッドで休ませてもらう。こんなにリアルな街にも「不法侵入罪」はないみたいで、誰も咎めない（これはた



あまり自由を満喫しすぎるとこういう目にあう

んなる皮肉)。

おおむね、圧政者がいても住民は善良である。が、しかし、巡回する邪悪なシャドーロードがいる。シャドーロードがいる都市に入ると憎しみの空気や臆病の気配を感じるので、そんなときはその都市はやりすごすのがいい。シャドーロードがいる都市の衛兵は私らを見かけると有無をいわず逮捕し、商人は金をちょろまかし、住人は会話しがてら何かを盗む。シャドーロードに捕まったら大変で、まず勝てない。しかし、悪いのは衛兵や住人ではないので、怒ってはいけない。

最後に、ウルティマVで遊ぶのに必要なものを書いておこう。

ひとつは根気である。なにせ、スーパー大河であるから。レベルアップも経験値をためるだけではだめで、ロード・ブリティッシュに会わねばならないのはウルティマIVと同じ。ただし、Vではロード・ブリティッシュは行方不明なのだ。そっと教えるとキャンプ中に亡霊が現れてレベルを上げてくれることがあるのだ。うーん、根気の野外キャンプである。

続いて異種世界、異種文化を楽しむ心である。優れたファンタジーはリアルな異文化を持った世界が描かれている。読者はその異文化を楽しむのである。劣ったファン



しゃべる馬の「エド」、じゃなくて「スミス」

タジーは現実世界をひきずった文化の上に成り立っているため、想像力をあまり要求されず読みやすいが、ファンタジーとしての魅力に欠ける。

さらに、筆記用具である。いつ、どこで役に立つかわからない膨大な情報。あっちへいったりこっちへいったり。メモが必要だ。経験を語ろう。ユーの街から別の都市へでかけると、ユーの街の誰それが知っているよといわれた。すぐにでも欲しい情報だったのでユーの街へ戻って尋ねた。すると、君は俺がそれを知っているということを誰から聞いたんだい？ といわれた。そんなことまでメモしてなかったのも、また危険な森を抜けて戻り、名前を確認し、またユーの街へ戻った。メモは重要。

それでもって、英和辞典である。なんといっても英語だ。たとえば、立て札や墓碑銘、看板にはルーン文字で書いてあるものがたくさんあるのだ。そして、それを表に従って解読すると英文が現れる。それを訳さねば何が書いてあるかわからないのだ。ほかにも英語がわかったほうがよい場面はある。このルーン文字を訳すのが面倒なことごとく。うーん。

では、みなさん、頑張ってください。ウルティマIVをやっていない人でも、終わっていない人でも大丈夫です。

総評だべさ

良くも悪くも、伝統と格式に守られた底の深さと指10本を駆使する操作性はウルティマである。誰の文句も許さない強さだ。ほとんどローリングストーンズのようなものだ。スターウォーズのようなものだ。

世の中にはちょっと聞いた分には耳に優しくノリやすくてヒットする歌謡曲や売れ線ロックと、ちょっと聞いただけでは異質で馴染めないけれど聴き込むほどに味の出る名作がある。ウルティマは後者のほうだ。ウルティマワールドに馴染むほど、味が出て、面倒だなんだと文句をいながらついつい大陸をさまよったり会話にうつつを抜かしてしまう。ストーンヘンジ4000年の歴史というか、ケルト人3000年の歴史というか、孔子の儒教2500年の歴史というか、デ

イズニーランド35年の歴史というようなそんな重みは重いのである。

5段階評価

ウルティマ度：★★★★★

ロード・ブリティッシュ度：★★★★★

非ドラクエ度：★

非イース度：★★★

道化師殺人事件度：★★

*

アメリカンジャーニー度：★★★★★

カリブの海賊度：★★★

ジャングルクルーズ度：★★★★★

シンデレラ城ミステリーツアー度：★★★★★

アリスのティーパーティー度：★

非スペースマウンテン度：★★★★

非スターツアーズ度：★★★

スプラッシュマウンテン度：まだ見たことナイ

THE SOFTOUCH

●プロミストランド



我が神が導きたもう 約束の地とは？

Yamada Junji

山田 純二

巷で人氣急上昇のポピュラスに、はやばやとシナリオ集が登場。西部劇編やブロックランド編などAmiga版からの移植5つと、イマジニアのオリジナル、江戸時代編の全6編が収録されている。まだ全面クリアしていない人もこれは見逃せないぞ。

5月に発売以降、巷で大好評のポピュラスにさっそく追加シナリオ集が登場。いままでは神と悪魔の対決という設定のみだったから、この朗報にはもろ手を挙げて喜んでしまったわけだ。

この追加シナリオ集には、インディアンと騎兵隊の戦い「西部劇編」、変な宇宙人同士の戦い「シリールランド編」、童心にかえってブロックとたわむれる「ブロックランド編」、ベルサイユのばら（ふっ古い！）を思い出す「フランス革命編」、未来世界での大手コンピュータメーカー同士の争い「ステーションナリーワールド編」、そしてなぜか武士と商人が戦うイマジニアのオリジナル「江戸時代編」と、6つのシナリオが含まれている。で、このバラエティ豊かなそれぞれのシナリオに合わせて、キャラクターデータもちゃんと変更されている。そのうえ、各面の設定条件やコンピュータ側の思考ルーチンにも変更が加えられている、オリジナルに比べると結構難しくなっている。というか、敵が強くなっているといったほうがいいな。

んでもって追加シナリオだから、プロミストランドを遊ぶには、と一ぜんポピュラスのディスクが必要になる。これを知らないとまさに宝の持ち腐れと化してしまうので注意すべし。

このプロミストランド、ルールや操作法、使える奇跡などはオリジナルのまま、特に変更はナシ。ただ、効果音も同じなのはちょっと残念。プレイしてみればわかるけど、各シナリオごとに特徴があるので、それにあった効果音が欲しくなってしまう。どれをとっても個性がつんつんしているとっても楽しいシナリオなので、戦いの音や沼に落ちたときの音がそれぞれ違っていたら、もっとよかったのに……。

この6つのシナリオのなかで、僕が気に入っているのは、江戸時代編での沼地。まるで、肥だめのような雰囲気をかもし出していて、落ちたらとっても臭そう。敵の民が落ちたときに、僕は今まで以上に、エクスタシーを感じてしまった（ん？ 危ないって？）。それでは、69面までプレイしたなかで、僕の気に入った（はまってしまった）、はたまた印象に残った3つのシナリオを紹介していきましょう。

そちも悪人よのう

ひとつ目は、江戸時代編。このシナリオは、ところどころに桜や松の木があって、なかなか日本情緒しているところが気に入ってしまった。特に面白いのが城の中庭。よ〜く見てみると松の木と玉砂利が敷いてあったりなんかして、細かいところまでやってくれるなあ、イマジニアさん、などとすっかり感心してしまった僕。まだ最初の面だからやりたい放題できるのをいいことに、新しいキャラクターの仕草を堪能しつつ、悪行の限りをつくしてしまった。

まず手始めに、必殺肥だめ攻撃！（うわあ、ディスプレイの向こうから臭ってきそう）もちろん、ただあちこちに沼を仕掛けるわけではなく、周辺に地震を起こして、



江戸時代編。桜も満開できれいだこと

いきなりですが、ポピュラス大会のお知らせ

夏休みにヒマを持て余している諸君、キミの「ポピュラスの腕を試すときがきたぞ！ なんてこんな企画が持ち上がったか」というと、なんでもポピュラスの原作者であるピーター・モリニュー氏がイマジニアのイキナはからいで8月25日に来日するそう。で、さすがは原作者、対戦ポピュラスにおいては未だ負けたことがないと豪語なさっているらしい。日本のポピュラスフリークともぜひ対戦を、てなわけで、あれよあれよという間にすっかりこの話が決まってしまったのである。

さてさて、この大会には7つのパソコン雑誌チームとイマジニアの計8チームが出場、おのの読者代表（イマジニアは違うらしい）をしたがえてこの大会に挑むわけだ。で、トーナメント形式で戦い、その8チームの優勝者がピー

ター氏と晴れて対戦、まさにポピュラスの王者決定戦というわけ。対戦期日は8月18日と26または、27日。まず18日に8チームの優勝者を決定、26または27日にピーター氏と対戦する予定。

そこで、だ、我がOh! Xでもゼツタイの自信と意欲のある読者代表を求めている。我こそはと思ったら、すぐさま官製ハガキを買いに走り、住所、氏名、電話番号、そんでもってこれがいちばん大切なワケだが、CONQUESTモードでの最高面数とそのパスワードを明記のうえ、Oh! X編集部「我こそはポピュラスの王者なり」係まで送ってほしい。応募の締め切りは8月5日（必着）。場合によっては、編集部で腕前を見せていただくのでウソや人から聞いたパスワードは書かないように。それでは、勇気あるポピュラスフリークの応募を待っているぞよ。



X68000用
イマジニア

5"2HD版 4,800円(税別)
☎03(343)8911

相手の民を引きずり出してから、沼を仕掛けるという極悪非道ぶり。そうすると、家から追い出された相手の民が、ボットンボットン、気持ちいいほどよく落ちる。

そうやってしばらく遊んでいると、相手の土地と自分の土地がつながるので、すかさず自分のシンボルであるまねき猫（相手のシンボルは「たぬき」だったりする）を移動して、民を誘導して敵地に突っ込ませる。当然、仕掛けた沼地は、地震と火山で潰しておく。でないと自分の仕掛けた罠に自分の民がはまってしまうという、間抜けなことになってしまうからね。

そのあとは、侍を作って相手の家に放火させてまわってネチネチと相手を攻撃させていったり、洪水を起こしてもう一度いじめ直そうかな、と思ったけど、あまりにも暗いので結局は最終戦争で勝負をつけて終わりにしてしまったのだった。

ぼくらの願いは世界征服だ！

さて、2つ目は子供の頃よく遊んだ覚えのある、ブロックの世界を使ったブロックランド編。マップが見づらいのが難点だが、これといって難しくはなかった。が、しかし53面！これがとにかく面倒だった。最終戦争を起こせないで、勝つためには相手を個別撃破していくしかなく、しかも騎士が作れない。なぜかという、圧倒的にこちらが有利になろうとも、相手を全滅させなければならないので、結局はシンボルを移動させ、リーダーをせっつきながら1つひとつ倒していくという、非常に非能率的な戦法を取らなくてはならないのだ！

攻撃しているときでも、相手はどんどんへんぴな場所に分散してしまうので、鬼ごっこよろしく追いかけて回させられる。そのうえ地面を盛り上げることしかできなくて、土地の整備が難しい。人が増えてくると当然のことながら全体の処理が重くなるため、マウスの誤操作がしょっちゅう起こる。せっかく苦勞して作り上げた平地が、ちょっとしたミスで水の泡になってしまったことが何度あったか。

この面はホント、これら悪条件のためにストレスが溜まってしまった。1時間も2時間もマウスをクリックしていると、肩もこるし目も疲れてくる。まあ、それだけに勝ったときには、すごくほっとしたけど。

哀愁のプログラマ

そして、3つ目のシナリオは46面のステーションナリーワールド編。僕がプロミストランドで初めて負けてしまったのがこの面。



まるでオモチャの国のようなブロックランド編



こっちはステーションナリーワールド編

日頃付き合ひの深いコンピュータ世界での戦いということで、このシナリオは結構はりきって遊ぶぞ！と思いきや……。

今までと同じようにシンボルを移動させながら、相手の土地を目指して進んでいたならば、しばらくして相手の火山攻撃。1発目のときは、わりと余裕たっぷりに、コンピュータも頑張っているなあ、と作られた山を削っていた。が、間髪入れずに2発目の火山攻撃を受けたときにや、マウスを握る手がピクリ。3発目には思わず、マジかよとつぶやき、4、5発目には目が座って、必死に復旧作業をする僕の姿があった。

すでに、連続の火山攻撃で泣きそうになっている状態に、さらに追い打ちをかけるように「フォン」と変な音が。思わず背筋がぞくとして、マップを捜し回ると、いた！ガチャピンナイト（このシナリオのナイトは、まるでボンキッキのガチャピンの頭に足を2本付けたようなやつで、その愛らしい顔とは裏腹に、領土を荒らし回ってくれる）。しばらくするともう1匹、さらにもう1匹と今度は、連続のナイト攻撃！もちろん、火山攻撃も休むことなく続いていて、結局はたび重なる敵の攻撃に耐えられず、負けてしまった。

あまりの悔しさにすぐさま再度チャレンジしたが、結果は同じく負け。しばらく呆然として、設定画面をながめていたら、“WATER IS FATAL”の1行に気づき、3度目の挑戦にして、ようやく勝つことが

できた。わかってしまえばなんのことはない。ナイトは海に沈めてしまえばよかったのだ。ここで初めて、プロミストランドが、難しいと実感した。

500面クリアした人はいるか？

このプロミストランド、それぞれのシナリオは見掛けはおちゃらけたパロディ。が、中身はなかなか手応えあり。それに初めからやり直すのが面倒臭ければ、オリジナルのパスワードが、そのまま使用できるので（サンプル版では）、自分が進んだ面から自由に遊ぶことも可能だ。

欠点としては、キャラクターを変えたことによりマップが見づらくなってしまっていること。ステーションナリーワールドは地面の盛り上がり方が滑らかにつながっているし、ブロックランドでは角張った地面なので、どこが窪みでどこが盛り上がっているか、慣れてくるまで区別が難しいカナ。

それにしても、オリジナルでさえ500面あるのに、さらに追加シナリオが出てしまって、単純に考えたら1000面。発売からしばらくたっているとはいえ、はたして全面クリアした人はいるか。スタッフでは、祝一平氏と西川善司氏の2人が、400面ちょっとのところで争っているようす。ほかには、200面、300面クリアの人がちらほら。しかしまだクリアした人はいないよう。全面クリアしたらどうなるか、気になってるんですけどね。

総評(天国は楽しい)

この追加シナリオ集は、それぞれのシナリオに合ったコミカルなキャラクターがわざわざと動き回り、見ているだけで楽しくなっています。以前、スペースハリアーで、キャラクターデータを書き換えたパロディ版があったのを覚えているでしょうか。あれはただのお笑いの世界でしたが、このプロミストランドはシナリオごとにそれぞれ因縁の対決を再現していてストーリーを感じさせてくれます。

さて難易度ですが、本文中でも述べたと思いますが、“6面から相手はナイトを作れるよう

になる”と、一言いえばわかると思います。キャラクター自体は可愛いくて愛敬もあるくせに、やることは手厳しい、まさに可愛さ余って憎さ100倍とはこのことです。先へ進むのは結構タイヘン。体力と時間のある方は、ぜひ挑戦してみてください。

総評(5点満点)

キャラクター	5
肥だめ	5
変身	4
シナリオ	4
難易度	5
やっぱり面白い	5

AFTER REVIEW

今月は、天下統一、ダウタウン熱血物語、あーくしゅの3つに加え、6月号の付録ディスクに収録したYet Another Columnを紹介しします。さあ夏休み、思う存分ゲームにひたれるときが来たぞ。この夏やりこんだゲームの感想をどんどん送ってちょ。

天下統一

▶最後の最後まで手が抜けない。最後までライバルといえる勢力が存在する。

岡山県・水口 仁郎(21)

▶私は日本史が好きです。

大阪府・加藤 弓弦(22)

▶現在のX68000のシミュレーションゲームでいちばん楽しめる。

徳島県・中沢 賢一(22)

▶戦国時代のようなすみをごとにシミュレートしているから。

広島県・平本 裕司(18)

▶コマンドはかんたんだがよくできている！

千葉県・根市 浩(27)

▶反射神経を必要としないし、自分の住んでいる国から統一にかかれる。

滋賀県・小池 清(42)

▶末長く遊べそうだから。

新潟県・保科 康広(20)

▶画面よし、音楽よし、内容よし。

京都府・可見 典明(17)

▶アルシスの移植と聞いただけで……。



熊本県・中村 巧(19)
▶ほかの機種で有名であったが、それがまたいちだんとパワーアップして登場。

鳥取県・安岡 正美(18)

▶戦いが城単位だから戦略的に自由度が高いのが、思ったより面白い。

北海道・近江 弘和(18)

▶戦国シミュレーションファンにはオススメ。

東京都・金子 博政(24)

▶かゆいところに手がとどく

北海道・釜蓋 実(19)

あのアルシスソフトが移植をして、システムソフトが発売した戦国シミュレーションとあって、発売される前から評判だったこのゲーム。フタを開けたらやっぱりこのとおり、の人気でした。フルマウスオペレーションもすることながら、やはりシンプルかつわかりやすい点が、ユーザーの共感を得たのでしょう。統一を目指していく手段も、国対国の争いではなく、1つひとつ城を攻略していくといったやり方なので、ゲームを進めていくうえで、非常にやりやすくてきているといえます。また、余計なものを排除したとはいえ、各々のグラフィックもなかなか見応えがあるものでした。しかも、評価版に比べて製品版はかなりスピードアップしているようです。

こういったシミュレーションものは、まず第一にコンセプトがしっかりしているかどうかにかかっています。シンプルでもいい、面白いものを、というその意気込みがひしひしと感じられ、プレイする側としても、うれしい作品でした。

X68000用 5"2HD版2枚組 9,800円(税別)
システムソフト ☎092(752)3902

発売中のソフト

★ギャラガ'88

電波新聞社の今度の新作は、ナムコの「ギャラガ'88」。「ギャラガ」というゲーム自体は1981年に発表され、未だにゲームセンターなどでちょくちょく見かけるが、このギャラガ'88は、1987年に発表されたそのリメイク版だ。自機を2連結・3連結させて、ギャラガ星人を心ゆくまで吹き飛ばしてちょうだい。X68000版には電波オリジナルのボーナスステージが追加される予定というから楽しみ。

X68000用 5"2HD版2枚組 8,200円
電波新聞社 ☎03(445)6111

新作情報

★遊撃王II

21世紀の近未来の空に展開する、最新鋭戦闘攻撃機「MI-C.A.D.O.II」型の活躍を描くフライトシミュレータ。ミッションブレイクモードのほかにはフライトシミュレートモードが用意され、まず訓練飛行・模擬戦闘でパイロットの腕を磨くことができる。MI-C.A.D.O.IIに慣れたらミッションブ

レイクモードに挑戦。迎撃、偵察、攻撃、護衛の中から任務を選ぶ。弾数や燃料を考慮し、みごと任務を遂行できれば昇格できる。目指せ、最高階級！ サイバースティックにも対応し、フライトシミュレータファンにはたまらない一作といえそう。

X68000版 5"2HD版 予価8,800円
システムソフト ☎092(752)3902

★Thrice

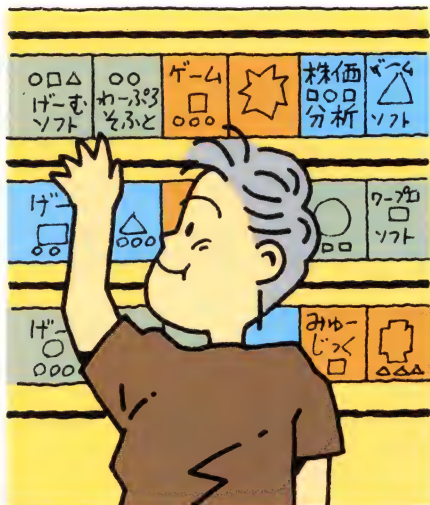
立て続けに新作を発表しているM.N.M. Software。今度はパズルゲームが登場だ。ブロックが上から降ってくるというのはお決まりだが、着地してから回す倒すひっくり返すの大騒ぎ。テトリスでもない、コラムでもない不思議な感覚のゲームだ。隠れフィーチャー、季節感のあるグラフィック、古代裕三氏のBGM、ビデオ機能に300名までのランキングと盛りだくさんに詰めこんだ、M.N.M.入魂の一作。

X68000用 5"2HD版 価格未定
M.N.M. Software ☎0423(60)3084

★サイバリオ

マニア垂涎のマト、あのタイトーのサイバリオンが家で遊べるようになるぞ。

メカニカルな龍をトラックボール(X68000版ではキーボードなども可)で操り、炎で敵も弾も振



ダウタウン熱血物語

▶お店へ入っているときのくにおやりきがかわいい。戦い方がいろいろあっていい。

長野県・山崎 芳照(15)

▶画面がどう考えてもX68000のものとは思えないが、やってみるとやみつきになる。

茨城県・関根 信男(17)

▶とんでもないマップさえなければ、最高なのになあ。

東京都・高見 創(19)

▶ファミコンの移植だからダメかなと思ったが、これが意外と面白いのよ!

高知県・井上 哲郎(24)

▶他人がどう言おうと私は好きや。

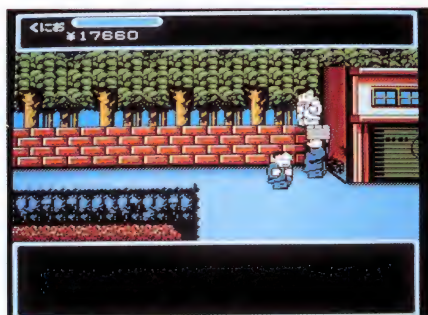
大阪府・渡辺 雅之(29)

たくさんのアイテム、殴る、蹴るなど日頃のうっぴんをはらすにはもってこいだったこのゲーム。やはり、そのあたりの単純さがよかったのかもしれない。マップはやや入り組んでいましたが、それがかえてこのゲームを面白くしたともいえるでしょう。

X68000用 5"2HD版2枚組 8,800円(税別)

シャープ

☎03(260)1161



あーくしゅ

▶ピクトのまじめさに対し、じえだのすっとばげた会話がすごくいい。

埼玉県・奥村 光雄(15)

▶じえだが二重人格者だから。おまけに言うところ、マウスカーソルはヤマトとウルトラマンとやじるしもあるぞ。

北海道・谷口 有香(21)

▶短時間で解けるのがいい。

東京都・合屋 琢(21)

このゲームに関してはカワイイ、とか面白いとかいったひとりで表せるような感想が多かったですね。いままでのウルフ・チームとはひと味違って、遊びの部分でできあがっているような感覚が、気負いを感じさせずかえってよかったのかもしれない。それに、なんといってもキャラクターがみんなかわいい。いずれにしても、ウルフ・チームは、こういったパロディものでも、シリアスものでも作れるという実力をみせつけた作品でした。

X68000用 5"2HD版3枚組 6,800円(税別)

ウルフ・チーム

☎03(5273)4795



Yet Another Column

▶Yet Another Columnにハマっています。テトリスの4段消しのときよりも、Yetの連続して消えていくときの気持ちのよさといったら、もう言葉では表せません。得点は3万点ちょっとなので、努力して4万点突破を目指すぞ!

静岡県・富永 恵隆(19)

▶な、なんなんだYetのあのスピードは(速くなったときのことだよ)。バカヤロウ、キーの反応が追いつかないくらい速く動かすんじゃないねー!

愛媛県・柳井 敏彦(31)

▶テトリスより熱中してしまった。ヘタに込み入ったゲームよりもシンプルで、なおかつ面白いのはこいつくらいだろうな。感謝であります。

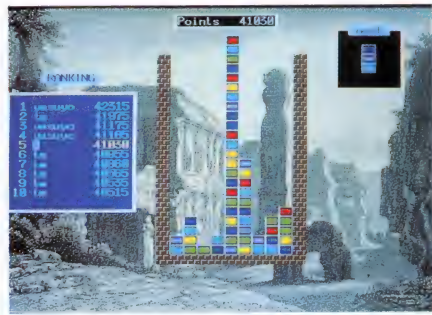
沖縄県・大城 久(18)

▶思いがけない連鎖反応が好き。

福岡県・村上 淳一(18)

まあ100号記念ということで、こんなのも今回は入れてみました。付録でつけたとはいえず、好評を得ているのは編集部としてもうれしい限りです。みんな、遊んでくれますか?

Oh!X1990年6月号付録ディスクに収録



り払ってこれまたメカニカルなボスキャラと対決する。実戦モードでは、プレイごとに独自のシナリオと独自のマップが作られ、ストーリー展開に従ってパワーアップしたり無敵化したりする。おまけにボスキャラのなかには、「ダライアス」のボスキャラも入っているとか。操作感覚に慣れるための練習モードもあるぞ。

移植はSPS、トラックボール対応とくればやがうえにも期待は高まる。今からトラックボールさばきを鍛えておこう。

X68000用 5"2HD版 価格未定

シャープ

☎03(260)1161

★ラグーン

言わずと知れた「ジェノサイド」のズームが放つファンタジーRPG「ラグーン」がいよいよ発売になるぞ。300年前。7人の魔導士が邪神を呼び出してしまったことがすべての発端となった。3人の命を犠牲にして邪神は封印されたものの、この一件は魔導士の間に決定的な影響をもたらした。邪神の力に魅入られ、その力を手に入れるべく「闇の皇子」を捜す魔導士ゼラー。そしてその闇から世界を守ろうと「ムーンブレードの勇者」を捜す魔導士マティアス。そして彼は少年ナセルとの決定的な邂逅を果たす。彼こそがムーンブレードの勇者なのだ……。

子供が泣きだすほどのデカイキャラと、ゲーマーが腰を抜かす激しいアクションに酔いしれてちょうだい。

X68000用

5"2HD版 8,800円

ズーム

☎011(613)0191

★幻獣鬼

古より、6つの魔界との接点「結界」に囲まれた王国ジタンの人々は魔物と戦う宿命にあった。しかし、ある日無能な魔導士が結界を破り、魔物が王国に攻めこんできてしまった。戦士レオン、魔導士リノ、忍者ルカの3人は、結界を封じる6つのロシュファの魂を奪い返すために旅立つ。MSX専用に開発された「アンデッドライン」が、X68000用にパワーアップしてリリースされる。プレイヤーは3人のキャラを自由に選び、好きなステージから攻略してゆく、キャラによって面のアイテムなどが微妙に変わるなど、数々の趣向を凝らしたT & Eの自信作だ。

X68000用

5"2HD版 価格未定

T & E SOFT

☎052(773)7770

★イメージファイト

つぎつぎとビデオゲームの移植が続いているなか、ついにシューティングゲームの真骨頂、イメージファイトが登場。20XX年、東西陣営の軍事競争のなか、突然西側のムーンベースが大爆発を

起こした。西側未確認の戦闘機によるものと判断した西側は、最新戦闘機OF-1を急ぎょ用意した。訓練飛行は完全ではないものの、コンピュータシミュレーション試験に合格した者は即、宇宙に飛び立っていった。

最初の5ステージがそのシミュレーション面になっており、平均90%の撃墜率をマークしたもののだけが実戦へ進むことができる。落第者は補習ステージ行きた。ボッドシュートやスピードチェンジ、特殊攻撃パーツを使いこなし、目指すはムーンベース内のマザーコンピュータだ!

X68000用

5"2HD版 価格未定

アイレム販売

☎06(535)4880

★ブルーサの復讐

反響を呼んだ「トリートン・ファイナル」の続編だ。剣と魔法を駆使する8方向多重スクロールのアクションだ。大魔王アレスターに侵略され、国を捨ててウオークの国にやってきたひとりの少女。彼女はウオークにくる途中、突然現れた悪魔により、船は難破し、彼女の兄は呪いをかけられ連れ去られてしまったという。勇者スタイルは、大魔王アレスターの持つムーン石で少女の兄の呪いを解くため旅立った。

X68000版

5"2HD版 価格未定

サイン・ソフト

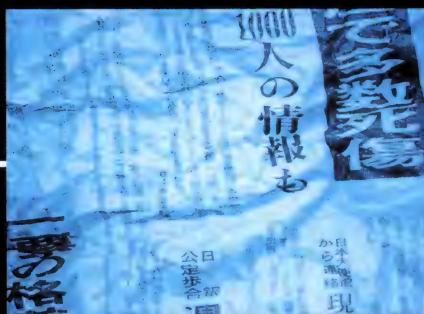
☎0794(31)7453

ADVANCED 2D GRAPHICS

X68000のX-BASICに初めて触ったときのことを思い出す。アナログRGBをサポートしたマシンのグラフィックは……と期待しつつLINEを引いて、表示されるギザギザした線にちょっぴり失望したものだった。

これまでのグラフィック特集ではどちらかといえば3D処理を主体にしていたように思う。これもX68000発売から比較的早期にZ'sSTAFF PRO-68Kが発売されたことが大きい。このツールはそれまでのパソコングラフィックの枠を超えた処理を実現した。そして名実ともにX68000の標準的グラフィックツールとなっている。

しかし、はや3年。内容はともかく、もはや新しいコンセプトのツールとはいえない。その他のツールもZ'sSTAFFに追いついていない。もっと違ったコンセプトに基づくツールができてもいいのではないかな？



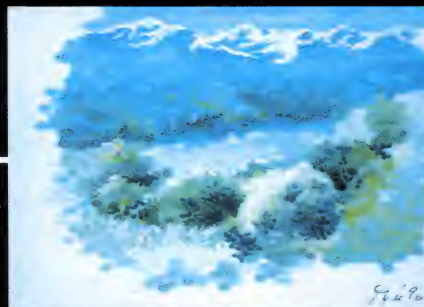
これをスクリーンとすると……



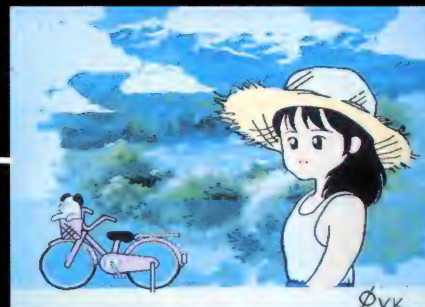
スクリーンとつきペイント



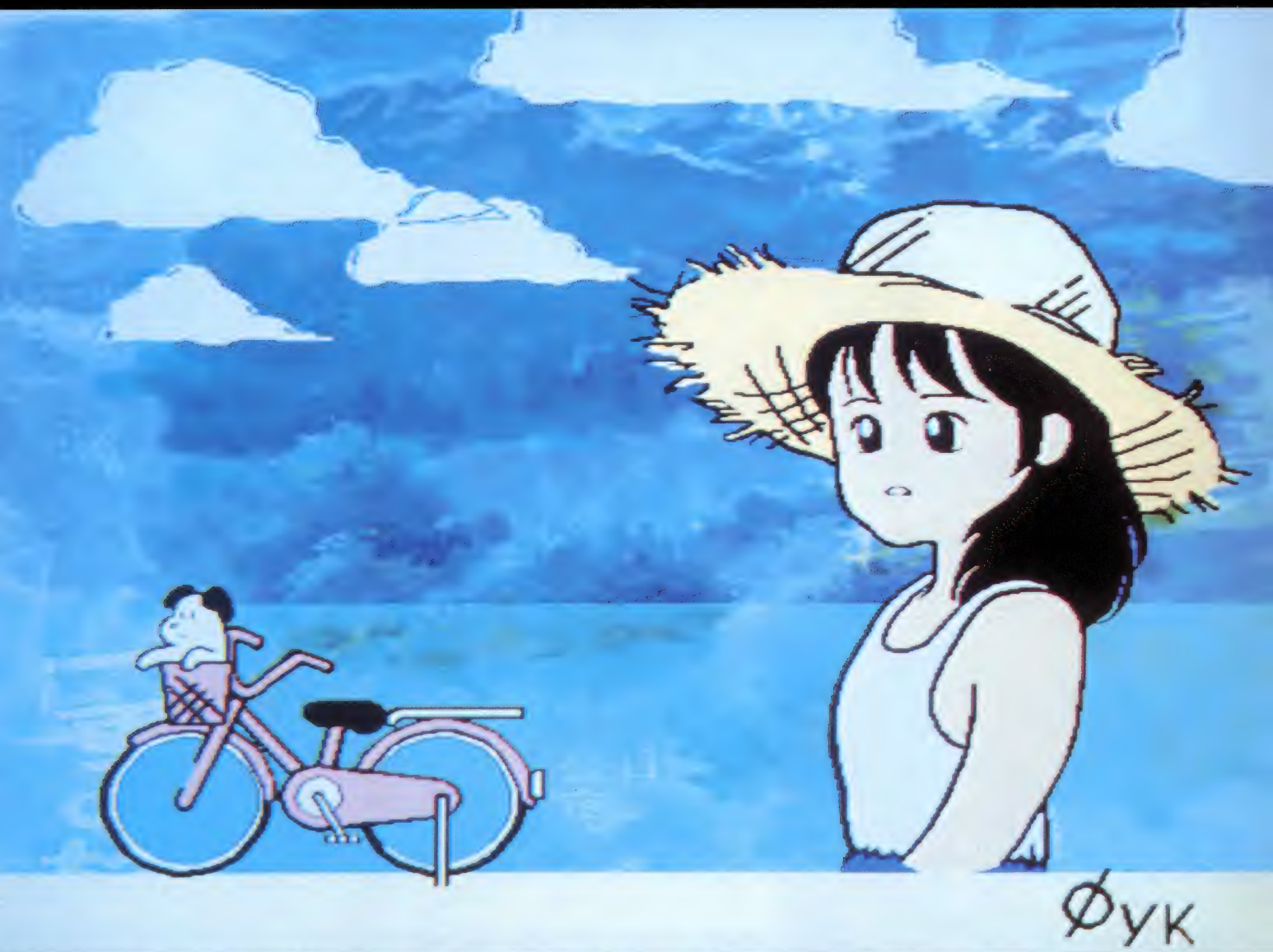
元絵



タイルとして登録し……



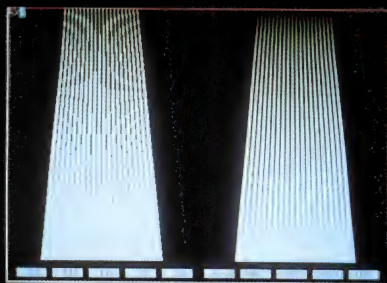
タイリングペイント



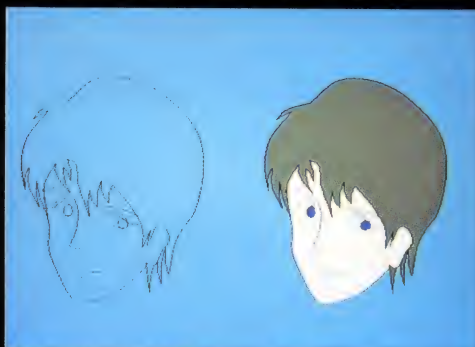
アンチエイリアシング対応スクリーン・トーン&タイリングつきペイントルーチンの応用例。タイリングペイントとはいってみればグラフィックパターンの連続張り付けだ。デジタルRGBでは多色表示のために使われていたが、アナログRGBではあまり使われない。メモリに余裕があれば張り付けるタイルの大きさに制限をつける必要はない。これはヘッダを書き換え、最大512×512ドットの画像をタイル登録できるようにした関数での実行例だ。空の部分にスクリーン・トーン（全画面分の新聞紙）とタイル（背景）をペイントした。ビデオなどのクロマキー合成に似ているが、アンチエイリアシング対応なので、マスキング不要で本当に塗りたい部分の隅々までペイントできる。スクリーン・トーンとは合成の比率を決定するもので、パターンさえ用意すればぼんやりとオーバーラップする画像や任意範囲の階調つきマスキングにも使える。

CONTENTS

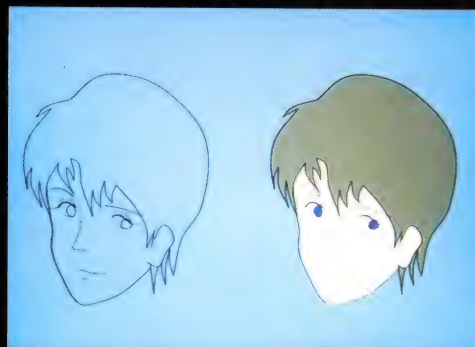
X68000用グラフィックツール紹介 あなたにあったグラフィックツール ……荻窪 圭	44
ギザギザのないグラフィック関数 アンチエイリアシングとは? ……丹 明彦	50
X-BASICによる画像処理 後処理によるジャギーの除去 ……中野修一	68
色数の補間と量子化 グラフィックを変換する ……鈴木康弘	72
4096色・8色変換 Zの画像をX1で ……亀田雅彦	77



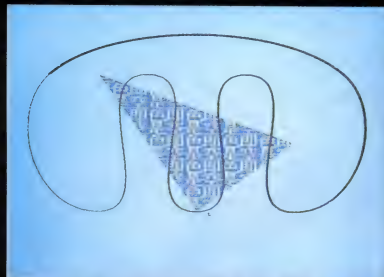
滑らかなラインを見よ



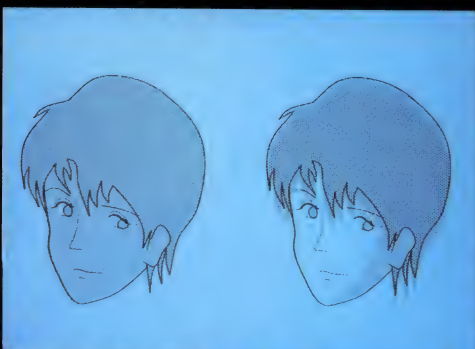
従来の関数による画像



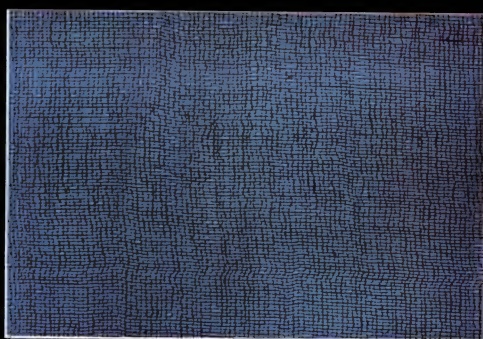
アンチエイリアシングされた画像



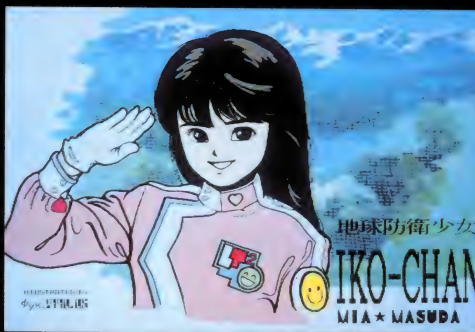
半透明のスキャンコンバージョン



スクリーントーンを使う



ガーゼを取り込み反転する



キャンバス地のような表現となる

丹明彦氏によるX-BASICで使えるアンチエイリアシング対応のグラフィック関数の使用例。滑らかなライン（ライン幅調整可能）、ベジェ曲線による滑らかな曲線、そしてタイリングとスクリーントーンに対応したスキャンコンバージョン（閉曲線領域の塗りつぶし）とペイントルーチン。すべてがアンチエイリアシングによる多階調の境界線に対応している。

これらの新しい関数群は単にいままでのBASICにあった関数の発展版としても使えるが、柔軟な思考で使い方を変えれば、さらに新しい可能性が見えてくるはずだ。すでに前ページで行った画面合成。機能が柔軟ならペイントでこういった処理までできてしまう。スキャナを使ってガーゼを取り込んだものをreverse()で反転し、tone_get()でスクリーントーンとして登録。左の写真のキャンバス地のような背景はこのようにして作られた。

2Dグラフィックもまだまだ面白い可能性を残している。

XROT 0によるグラフィックの回転

特集の記事ではないが、読者投稿によるグラフィック回転プログラムの実行例。短いプログラムでしかもかなり高速。サンプルプログラムはキー操作により拡大縮小自由自在でぐるぐる回転する。画面下の領域に画面の内容が再帰的に反映されているのも面白い。デモやゲームの特殊効果はもちろん、グラフィックツールの一部として使っても面白い機能だ。



これが元の画像



回転後。下に再帰している部分が見える



元の画像 (640×400)



拡大図 タイリングが見える



512ドットで変換



640ドットで変換

PC-9801のアナログ16色モード用に描かれた画像をX68000の65536色モードに変換した例。もちろんアナログパレットにも対応し、タイリングパターン部分を単色のベタ塗りに変換する。画面サイズを補正して512×512ドットの全画面対応に変換した場合は輪郭線の一部が切られてしまっている。横640ドットにしたものはプロポーションが原画と異なっているが、ドット構成は正確に変換していることがわかる。CRTCをいじれば横768ドットのまま正方形に近いピクセルで65536色を出すことも不可能ではない(自由研究)。タイリングが単色に置き換えられるため、65536色化されたもののほうが元ファイルより圧縮がきく場合も多い(PIC.R)。なお、原画は森林林橋氏によるもの。



タイリングが消えた



これが65536色

65536色の画像をできるだけ原画に忠実に256色モードに変換した例。オーダーディザ法を使ったものと菜野雅彦氏のアルゴリズムを応用して多色化したもの。よく見ないとわからないが比較的規則的なパターンになるディザ法と、かなりランダムなパターンになる菜野氏のアルゴリズムの違いがパターンになって表れている。階調表現は菜野式のほうが自然に思えるが、もともと白黒2色用なためか、隣接するドットの明暗差が激しく出るのがやや気になる。多色用にアルゴリズムを改善できるのではと思う。誰か挑戦してみてください。



オーダーディザ法



これは菜野式

X68000用グラフィックツール紹介

あなたにあったグラフィックツール

Ogikubo Kei 荻窪 圭

よくこーゆーことをいうやつがいる。
「それで、なにが面白いの？」
ノートに落書きしてるのを見ても絶対そ
んなことを聞いてきたりはしない。

「で、さあ、それって、役に立つの？」
役に立たなきゃいけないときたか。
「プリンタで打ち出したりできるの？」
できねえよ（画面と同じようにはね）。
やっと、そいつのいうことがわかった。
紙に描いてあったり、ビデオで見られたり
しない絵は価値がないというのだ。

グラフィックを描いて遊ぶなんていうの
は、コンピュータはなにか役に立つもの、
と信じている善良な市民にとって信じられ
ないことらしい。そう考えてみると、グラ
フィックツールで遊ぶなんてのは、かなり
贅沢な道楽のようなのだ。道楽万歳。

* * *

目の前に山があるからといって別に登り
たいとは思わないが、目の前にあるのが紙
とペンだったりするととにかく描きたくな
り、楽器だったりするととにかく音を出したくな
る。誰も文句はいうまい。目の前にあるの
がポピュラスだったりすると締め切りも忘
れて沼を作りたくなるというバリエーショ
ンもあるぞ。

それでもって、目の前のパソコンにFM
音源やAD PCMが乗っていれば鳴らした
くなるし、65536色出るとわかれば色を出し
たくなる。ポップアップハンドルがあれば
持ち歩きたくなるし、ディスクがオートイ
ジェクトならゲットイン/ゲットアウトし
たくなる、ってなもんだ。それが人情とい
うもので、それが楽しいわけである。

そういったわけで、お絵描きソフト集合
である。X68000はワープロよりもグラフィ
ックツールが多いパソコンとして有名だが、
グラフィックツールといってもいくつも転
がっているわけで、片っ端からあさってい
たら私の身がもたない。

で、今月は2次元のお絵描きソフトであ
る。2次元のお絵描きというのはつまり、
CRTに投影されているグラフィックVRAM

をべたべたとデータで埋めていくことを目
的とした作業のことだ。これがグラフィッ
ク画面で遊ぶ基本。

グラフィックモードへの対応

X68000の場合、ご存じのとおり、グラ
フィックモードをたくさん持っている。

まず1024×1024（表示画面は768×512）
の16色。ドットが小さくて、1ドットの縦
横比がほぼ1:1である。

続いて、一番メジャーな、512×512の65536
色。1ドットを16ビットで表現しようという
贅沢さて、512KバイトのグラフィックRAM
がたったひとつの画面に収まってしまうと
いう恐ろしいモードである。

さらに、意外とおいしい512×512の256
色。1ドットを8ビットで表すわけで、2画
面分持てる。さらに、512×512の16色（4
画面だ）。

その下に、256×256モードがそれぞれあ
って、このモードは1画面当たりの情報量
が少ないため、高速な処理に向いている。
シューティングゲームに多いモードだった
りする。

とまあ、こんなにあるわけで（ほかにも
いろいろ隠れてたりするけれど）、すべての
モードに対応しているグラフィックツール
なんてない、のだ。

順番に見ていくと、まず756×512ドット
の16色モード！ に該当するグラフィック
ツールは、なし、である。PDSにもあると
いう話は聞かない。SX-WINDOWはこの
モードのグラフィックをサポートしている
ので、そのうち出てくるかもしれないが、
いまのところ、ない。

これはこれでけっこう綺麗な絵を描けた
り、文字を埋め込むには向いているのであ
ってもいいと思うんだが、ないなあ。SX-
WINDOWがこのモードだから、もしかし
たら、そのうち、マックペイントの玩具み
たいなのがでてくるかもしれない。また、
PC-9801のグラフィックのちょっと大きい

これまでにX68000用として発売されてい
るグラフィックツールを集めてみました。
それぞれの個性や使い勝手について独断と
偏見を交えて試用レポートをまとめました。
皆さんのツール選びの参考になるでしょ
うか？ それではサンプルは電腦絵師の福原
徹でお送りします。

やつだと思えば、また違ったものが出てく
る可能性もある。

さて、512×512ドットの65536色、といえ
ば、Z'sSTAFFと、G68Kである。X68000
で一番有名なモードだ。このモードにも欠
点があって、それは「メモリをたくさん食
う」とか、「ファイルが大きくなる」だ。も
ちろん自然画を扱おうと思ったら、このモ
ードでないと困るが、自然画は圧縮しづら
いのでデータの保存が大変。てなわけで
MOディスク万歳。

512×512ドットの256色。実のところ、手
描きであれば、このモードで十分な気がす
る。そこに気がついたのがサン・ミュージ
カル・サービスであって、マジックパレ
ットという軽快な異色グラフィックツールを
出してきた。開発がサン・ミュージカル・
サービス、発売がミュージカル・プランと
いう音楽業界コンビのグラフィックツール
である。

それから、ウルフ・チームのPRISMもこ
のモードが中心だ。一応こいつは256×256
モードや65536色モードなどもサポートし
ているが、メインは256色。ゲーム屋さんら
しい構成である。

ゲームソフトメーカーというのは、つい
ついグラフィックツールを出したくなるよ
うで（そりゃあ、社内で使うために作った
ものがあるはずだし）、ザイン・ソフトから
も予定されているようだが、間に合わな
かったのでとりあえずこの4本だ。テラツ
ォなんてのもあるが、あれはスプライト系
なので今回ははずす。

X68000の主なグラフィックツールはこ
の4つだ。256×256ドットモードのときは、
512×512モードの左上4分の1を使えばい
いわけだから、問題はない。なかにはちゃ
んと256×256モードをサポートするツール
もある。

画像フォーマット

続いて、とにかくにもグラフィックツ

ールを使ううえで問題となるのが画像データのフォーマットであった。いくらたくさんツールがあっても、それぞれみんな勝手気ままなフォーマットでセーブされたら、たまったもんじゃない！ ってことは、有史以前からいられていた。クスコーの壁画にも書いてあったほどのだ。

X68000の場合、非常に幸運なことに、3つの標準的なフォーマットがある。そのうちの2つはたいいてのグラフィックツールがサポートするというラッキーな結果だ。

第1がGL3 (65536色モード時) フォーマットである。ベタフォーマットともいう。X-BASICのIMG_LOAD, IMG_SAVE関数で読み書きできるフォーマットであって、X68000ユーザーなら誰でもこれでセーブされたグラフィックを読むことができる。

ちなみに、256色モードではGM3, 16色モードではGS3, 256×256ドットモードでは3番目の数字が0になる。

この方式の面白いところは、セーブされた画像のモードをファイルの拡張子で区別していること。ファイルには画像データしか入っていないのだ。つまり、どのモードでセーブしたかがデータを見ただけではわからないのだ。私はこういうのはアナーキーで好きだが、無秩序で嫌いだという人もいるかもしれない。この方式をサポートしていないのは、上の4つのうち、Z'sSTAFFとPRISM.G68Kにいたっては、GL3フォーマットを標準フォーマットに採用している。

ちなみに、この形式はもちろんデータ圧縮をしないため、512×512の65536色だと1枚セーブするたびに512Kバイトの磁性面を消費する。ディスク1枚に絵が2枚しか入らないわけだ。

第2が、ZIMファイルである。これは、とにかく権威のZ'sSTAFFである。X68000用で初めてのグラフィックツールで、あまりにメジャーなため、あとから出したソフトはたいいてこのファイルを読む機能なり自分のソフトのフォーマットに変換するツールなりをつけることとなった。

圧縮形式と非圧縮形式があり、たいいて非圧縮形式をさす。ZIMファイルはX68000に向いているかという、そうではないという意見が大半を占めていて、評判はあまりよくない。

Z'sSTAFF (当たり前だ) のほか、G68K, PRISMがサポートしている。

3番目がPIC形式。PIC.RというPDS(正しくはフリーウェア)の画像データ圧縮・展開ツールの形式だ。圧縮効率が高いのが好まれるところ。でも、PDSなもので、市

販のソフトで対応しているものはなかったりする。自然画を使うのでないならば、とても有効だ。

しかし、どのグラフィックツールも、画面にロードした絵を消さないで起動する方法があるので、ファイルコンバートよりも、こいつを使ったほうが楽だったりする。

では、ひとつずつ簡単にレビューしていこう。

Z'sSTAFF PRO-68K

とにかく、あまりにも有名。PC-9801用のZ'sSTAFF KID-98やらX1turboZについてきたZ'sSTAFF Zからの伝統芸は衰えるきざしなし。伝統の重みはX68000にまで及び、PC-9801なんかと互換性のあるファイルフォーマットを持ち込む(ZIMファイルと呼ばれる)という荒技に出たが、それが唯一の欠点らしい欠点である。

これについては、恐怖の常駐ソフトPIC FILERなるPDS (正しくはフリーウェア) が電腦倶楽部に掲載され、ひとつのマニアックな解決を見せている。これはPIC形式ファイルのロード/セーブをZ'sSTAFF上から行うものだ。

次のバージョンではPICとはいわないが、GL3形式のロード/セーブくらいはサポートがほしい。

メニューは画面一杯開いてまだ余るくらいたくさん開ける。下がPICFILERを使ったところ。本体のみでも自由変形に色変換と機能は尽きない。強いて欠点をいえば、マスクのみのセーブができない、2枚の絵を重ねる機能などがないというところか



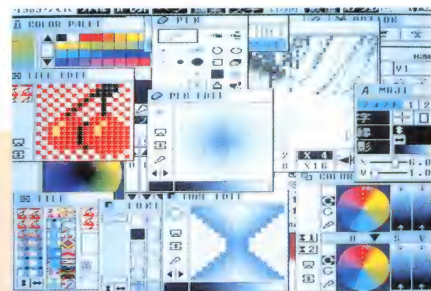
Z'sSTAFFによる作画例

操作の基本は、プルダウン風のメニュー。メニューバー上のメニュー、ファイル、パレット、ペン、編集、文字、印刷、数値、オプションの8つから必要なものをクリックすると、ぼよんとウィンドウが開く。その気になれば、描いたグラフィックが全部隠れるほどウィンドウが開きまくる。

グラデーション、トーン、タイル、にじむ色、自由なペン先、スプライン曲線などお絵描きの機能はやたら豊富。

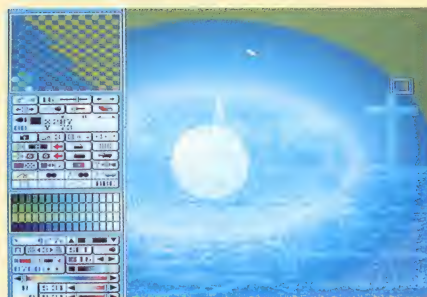
特にそのグラデーションパワーはライン、ペイント、ボックスフィルや閉曲線ペイントなどいつでもどこでも使え、誰でも描ける富士山とか誰でも描ける円柱などの技を作り上げた。

ペン先やブラシだけでなく、ポップで派手なタイルやトーンなどほとんどのものが編集可能で、特においしいのが濃淡の調節である。ペイントやカラーチェンジも、指定範囲内の色に対して行えるので、各種効





鉛筆画っぽいイメージを目指してみた。セピア調でパレットを統一し、極細ペンを使ってマウスでこりこり……。256色512×512モード固定ながら、なかなか多彩な機能があって使い慣れれば相当器用な絵も描けるのではないかなと思わせる。ほとんどの機能がメインウィンドウに収まっているのでわずらわしさがない。消しゴムもいい。



果が狙える。

編集機能も任意矩形の回転・変形・拡大・縮小、任意曲線内のムーブ・コピー。気になるのは、ムーブしたあとに残る白い跡。背景色が白になっているためだ。

外部入力についてもスキャナからカラーイメージユニットまで対応している。バージョン2からはJIS第1水準のみだが、明朝体とゴシック体のアウトラインフォントもサポートされ、X68000ではどのソフトよりもきれいな漢字が書ける。グラフィックに淡色のグラデーションアウトラインフォント文字を入れたら、実に気持ちがいい。

おっと、忘れていたが、一部では致命的ともいわれた「プロテクトモジュール」によるコピープロテクトは、現在発売しているものにはなくなっている。買ったらずい

ていなかったのが驚いた。よいことだ。プロテクトモジュール付きのバージョン2.0を買ってしまった人は残念でした、と。

欠点といえば、プログラムがでかいので、メインメモリが2Mバイトないとアンドゥ機能が使えないことと、アウトラインフォントを使おうと思ったら、ハードディスクがないと大変だということくらいだろう。

512Kバイトの広大なメモリをアンドゥするのは大変だとは思いますが、2MバイトでもRAMディスクをとったり、変なものを常駐させたりしていると駄目である。フリーエリアが1.5Mバイトくらいあれば大丈夫だ。

それから、右ボタンで途中の作業をキャンセルするのだが、「ひとつ前の状態に戻るのではなく、その機能自体がキャンセルされてしまう」のはいただけない。

お絵描きツールのユーザーインタフェイス

いつか祝センセが書いてましたが、ユーザーインタフェイスというものは、たとえ操作しやすくなったとしても、それが古いタイプのものよりも格段にメリットがない場合、人はわざわざ新しいほうに移らないものだそうです。

X68000の場合、最初にZ'sSTAFF PRO-68Kという強力なツールが発売されていたから、後発のソフトは信者獲得には辛いものがあつたろうと思われまふ。僕自身がZ'sSTAFFの虜となっているので、今回の寸評もそこからの視点を中心に書いてしまっているのではないかと少々不安もあつたりします。

が、正直なところ、僕はZ'sSTAFFのようなウィンドシステムは好きではないのです。「下が見えなくて邪魔」なのが主な理由です（これは開発者も感じたらしく、Ver.2ではウィンドウが若干小さく変更されていました）。

グラフィックツールにウィンドシステムはあわない気がします。かといって、ウィンドウ以外に機能を使いやすく配置する方法っていうのが、まだわかっていないんですね。描画画面を小さくして周りに配置してしまうのも手でしょうけど、画面を有効に使えなくて悲しいし……。いい方法はないでしょうか。（T.F.）

まあ、どっちにしろ、機能と表現力ではまだ他の追随を許さない。Z'sSTAFFの天下はまだ続きそう。

マジックパレット

256色モードに目をつけただけでなく、ペインティングソフトとしてのインタフェイス構造も新しい。ファイル入出力用メニュー画面。ワープロやエディタみたいなカット&ペースト。アンドゥ用メモリ。メインメモリを2Mバイト積んでいけば、チャイルドプロセスでコマンドシェルを起動できたり、描画画面を3面持てたり。

円のグラデーション（外周から中心へのグラデーション）が派手なおかげで、ほかにもあるユニークな機能は見落とされがちだが、アンドゥ用メモリから任意の形で前のデータを切り出せるとか、カット&コピーバッファも編集できるとか、パレットコード&H00を透明色に固定し、背景の基本を透明色にしていること（画面の重ね合わせに便利）などなど。

特に背景が透明色だというのは嬉しい。どこでどう間違ったのか、絵は白い画面に描くもの、といった重力に魂を引かれたソフトが多いからだ。

まず、ファイル入出力モードで立ち上がる。3画面+コピーバッファ、そして6つのパレットに入れたいファイルがあつたら読み込むのである。終了時もこの画面に出て、セーブするなりする。デザインはともよい。

そこからコマンドシェルを起動することもできる。drawを選ぶと縦長でかくてデザインを優先したようなポップなウィンドウが現れる。アイコンがたくさん並んでいて、カラフル。ウィンドウは3つに分かれており、上1/3が描いたりコピーしたりするもの。まん中がパレット。その下がパレット関係の処理。

たとえば、グラデーションバーの両端に



絵の一部をペンとして使う

色をセットして、そのあいだの色の变化パターンをいくつにするか決める。それでもって、パレット上のそのグラデーションをセットしたいところへ置くと、ずらっとグラデーションした色がパレットに置かれるのである。マジックパレットというグラデーションは、あるパレット番号からあるパレット番号への色の並びにすぎないので(中身の色はなんでもいい)、赤黒青緑といった4段階グラデーションもできるし、虹も描ける。

処理の基本は前にも書いたが、カット&ペーストである。任意領域をカットしてバッファへ移し、それを任意の位置へペーストする。バッファを編集したりできるし、透明色を背景にしておくと、重ね合わせが簡単にできる。

その代わり、縮小・回転・変形処理が任意の領域に対してできない。回転や縮小をしたときは、対象のものだけをほかの画面へ持っていき、画面全体の256×256の画面に対して行ってから、戻すといった作業が必要で、複雑な絵を描こうと思ったら、まめにパーツをセーブするのがいいだろう。あと、トーン処理も面倒だ。

アンドゥ処理はユニーク。アンドゥはアンドゥ用画面メモリから戻されるのだが、そのメモリへのデータ格納は手動なのだ。で、面白いことに、消しゴムを使って画面を消すと、その下にはアンドゥ用メモリの画像が現れるのであった。アンドゥというより、いろんな画面効果に使えそうだ。

無理をいえば、任意のパレットを使ったカラーイメージユニットからの取り込みか、もっと上手な65536→256色変換がほしい。メニューやコピー時の領域が画面内に制限されているので画面が狭く、ちょっと不便なものも惜しいところだ。

なんだかんたいたって、Z'sSTAFFの影響を免れないソフトが多いなか、こいつだけは違う。非常にポップで軽く遊ぶには最適だ。

16色モードでは画面上の絵をスプライトデータに落とすことも可能だし、起動時にSキーを押しながら立ち上げると直前に走っていたゲームなどのスプライトデータとスプライトパレットを読み込んでくれるのでスプライトエディタとしても使える。

おまけで、マジックパレットのデータをBASICで使うための関数やBASICプログラムのサンプルがついてきて、とても便利である。ついでに、Cのライブラリもあればコンパイルできてよかったのに。オートデモもある。



太めのペンでベタベタと描いてみた。油絵調に見えるかな？(河○純子ちゃんがモデル)マウスボタンの左右に色を設定でき細かい修正に便利。マウスの反応速度を調整できるのもいい。特殊効果に弱いのとスキャナ・プリンタに対応していないのが辛い。

PRISM

こいつはZ'sSTAFFの影響を逃れられなかった。最初から大樹の陰にいたのかも知れない。

ウリは、2色から65536色まで、256ドットから512ドットまで対応した多彩なモードと、アニメーション機能。さすがウルフ・チームである(そーいえば、昔侍ジャイアンツにウルフチーフって選手がいたなあ)。

しかし、なんといっても、円が描けないとかグラデーションペイントができないとか文字入力がないとかペン先やブラシの編集もできないとかカラーイメージユニットもイメージスキャナも使えないといった事情にはなにか深いわけでも……と考えてしまう。

その他の操作性は遅いZ'sSTAFFという感じだ。ウィンドウデザインも似ている。

ウリはやはりアニメーション機能が。画面上の任意の矩形をたくさん切り出して、連続して見せてアニメーションしてしまうという機能だ。まずマウスで1コマの大きさを決め、15コマまで任意の位置を切りとって並べる。1/60秒単位で1コマの時間を指定できるから、サブリミナル効果測定テストなんかもできて面白いぞ。



どのタイミングが一番いいかテストして、学園祭では売り上げ倍増だ！(そんなにうまくはいくもんか)

画面を2画面まで持てるので片方を背景に使うとかすれば、なかなか、このソフトの意図も見えてくるかもしれない。

ゲームでは特殊な画面モードを使ったりするためか、ふつうのグラフィックツールではサポートしないような512×256ドットモードなどや16色モードなどにも使えるが、その半面、どのモードでもできるような機能しかついていないのが残念だ。とりあえず、どのモードでも絵は描けることを特徴としている。

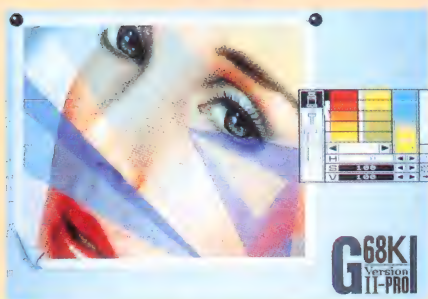
256色モードでアニメーションして遊びたい人は、マジックパレットとPRISMの2つを買って、マジックパレットで描いてPRISMで動かす、ってのもいいかもしれない。定価ベースでは、この2つを買ってもまだZ'sSTAFF PRO-68Kより安いのだ。



PRISMの使用例。油絵調を狙ってみた



手元にあった鉛筆の落書きを、ハンディスキャナで取り込んでエディットした。水彩画を意識してみたが……。ウィンドウ操作が比較的速いのと、カラーチェンジや閉曲線コピーなど編集機能が多く揃っているのがよい。ただし出力ファイルの形式がGL3なのが少々不満。ルーペは画面一杯に拡大し、そのままエディットできる。



G68K II Version 2.0-PRO

バージョン1では、日本初のBGMつきグラフィックツールという快挙を成し遂げてくれたG68Kであるが、バージョン2ではおとなしい作り（というかまともな作り）になっている。

立ち上げて驚くのが、真っ白な画面にポツンと十字カーソルがあるだけのまぶしい画面。メニューバーからメニュー選択するプルダウン式ではなく、その都度右ボタンでメニューを開いていくポップアップ式なのだ。

たかがツールされどツール

CGコンテストの審査などでよくいわれることですが、応募されてくるものに「こんな機能を使ってみました」みたいな作品が結構多いのです。ツールの豊富な機能を使うのはいいのです。でも、それに振り回されて自分の表現したいものがあやふやになっては駄目ですね。作品はツールの機能紹介ではないのですから、饒舌すぎないオリジナリティのある作品を描いてもらいたいです。

それには自分にあったツールを深すことも必要でしょうし、最終的には自分自身で組んだ、自分のためのお絵描きツールを使うのがベストなんでしょうね。

昔、(PC-9801の話だけけど) Z'sSTAFFと並んで有名だったグラフィックツールにシステムソフトのアートマスターというのがあった。このアートマスターもポップアップメニューで、アイコンやらメニュー構造などが非常に似ている。要は慣れの問題で、開いたウィンドウがうっとうしいという人もいれば、いちいち右ボタンでウィンドウを開くのがうっとうしいという人もいる。

機能的にはグラデーションペイントがないくらいで、普通。

パレットにタイル模様もセットできたりとか、マスク機能が使いやすいといった長所もある。使い勝手の差は、ポップアップメニューが馴染むか否かだろう。

独自のファイル構造や圧縮方式を持って

おらず、データはすべてGL3形式というのが素直といえば素直でよい。

Z'sSTAFFをよほど意識しているらしく、Z'sSTAFFの非圧縮ZIM形式とGL3形式の相互ファイル変換が可能となっている。

機能的にはZ'sSTAFFと比べるのがかわいそうだが、価格が半分以下であること、Z'sSTAFFより少ないメモリで動くといった面もあり、一概にはいえない。

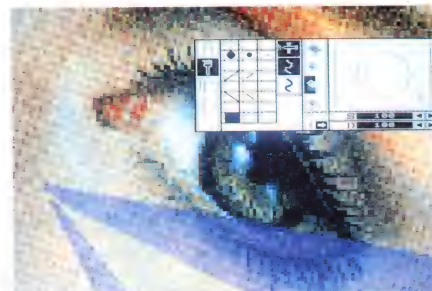
綺麗なサンプルとオートデモあり。

まだまだ、先はあるのである

X68000のグラフィックツールといえば、多くのユーザーや開発者がZ'sSTAFFを基準にしてきた。それはそれでいいとして、Z'sSTAFFが完璧なソフトか？ というところ、決してそんなことはないのである。そのひとつの例をマジックパレットが証明したわけだが、まだまだいろいろ便利な機能はあるはずである。マジックパレットだって、早く次のバージョンを！ てな感じだ。画像取り込みの柔軟さと、任意矩形の変形はほしいところ。グラデーションなんて簡単に綺麗な効果が出せるだけで乱用すると見苦しいだけだし。

えっと、コンピュータを使って絵を描くことの意義を考えてもらいたい。絵心のあつた人がペンをマウスに持ち替えて、ああ、よかったね、という時代は過ぎ去った。わざわざマウスを持たせるのだから、結局ペンで絵の描ける人でないといひこさせない、というのは変である。Z'sTRIPHONYやC-TRACEなどはデッサン力がなくても、センスと根性と待つだけの暇とちょっとした頭があれば誰でも使えるものだった。2Dグラフィックツールも、そんなものが出てきてもいいではないか。

たとえば、遠近法矩形や、始点と終点で太さの変えられるペン。任意の方向へのグラデーション。多彩なアンドゥや下書きプレーン（メモリの関係で大変だろうけど）。別に65536色でなくとも、32768色でも16384色でもいいので、そういった支援機能を充



メニューは邪魔にならないポップアップ式

実させるのも手だろう。だいたいにして、1万色あればたいいこと足りるはずだ。

あと、いろいろと難しいだろうけれど、PICファイルのサポートもあると助かる。

それでもって、一番ほしいのが、キーボードマクロと自動実行マクロと数値関数(三角関数など2次曲線の描けるもの)だ。

たとえば、規則的な図形をいくつもずらして描きたい、とか、ちょっと三角関数を使った線がほしい、とか、さっき描いたやつをもう一度描きたいなんてときはあるはずだ。マウスで行った一定の動作を覚えておいて、任意の点からそれを行えるというのがキーボード(?)マクロ。メニューから関数を選び、パラメータや軸の単位を与えて、マウスで指定した範囲に指定した色で指定した太さの曲線を描いてくれるのが関数機能。それでもって、プログラムウィンドウが開いて、ちょこちょこと簡単なプログラムを組むと、それを実行して図形を描いてくれるマクロ。

つまり、BASICでちょこちょこと描ける程度のをグラフィックツール上でやれたら面白いだろうな、と、思うわけだ。ついでに画面に適当に描いた自由曲線をフリーエ級数展開して三角関数の組み合わせに直してくれる機能、なんてのはあったら楽しいけど、そこまではうまい。

* * *

Z'sSTAFFを買ったはいいけれど、白い画面を前にして、グラデーションの空を描いたまま石になってしまった人や、マジックパレットを買ったはいいけれど、グラデ球をたくさん描いたまま凍ってしまった人も多いと思う。ときには石になって自分の才能に謙虚になるのもいいけれど、そうでない気楽なグラフィックだって実現できるはずなのである。

いま、思ったのだが、ドローイング系のグラフィックツール(パーツなんかを組み合わせて絵を作るツール)がない。2次元のグラフィックツールにはドローイング系のツールとペイント系のツールがあって、ここで紹介したのは全部ペイント系のツールだ。どうしてだろう。今度よく考えてみることにしよう。

- Z'sSTAFF PRO-68K [Ver.2.0] 58,000円
ツァイト ☎03(299)0461
- マジックパレット 19,800円
ミュージカル・プラン ☎03(401)2751
- G68K version II -PRO 22,000円
SYSTEM HOUSE OH! ☎075(502)2972
- PRISM 68K 38,000円
ウルフ・チーム ☎03(5273)4795

(価格はすべて税別)

機能比較一覧表

		Z'sSTAFF PRO-68K	マジックパレット	G68K version II -PRO	PRISM
画面モード	メイン対応	512×512,65536	512×512×256	512×512,65536	512×512,256 512×512 256×512 512×256 256×256 (2,4,8,16,64,256,65536色)
ファイル形式	ZIM	○		○(非圧縮)	○(非圧縮)
	GL3		○	○	—
	独自		○		○
カラー	グラデーション	縦/横	縦/横/円	—	—
	スポイト	○(2種類)	○	○	○
	タイル	○	○	○	△
	トーン	○	△	○	—
	混ぜ合わせ	○	—	—	—
	濃淡	○	—	○	—
	透明色機能	—	○	—	○
ペン	太さ	19種類	7種類	24種類	19種類
	ペン先編集	○	△	—	—
	BOX/FILL	○/○	○/○	○/○	○/○
	円/FILL	○/—	○/○	○/○	—/—
	楕円/FILL	○/—	○/○	○/○	—/—
	扇/FILL	○/○	—/—	○/○	—/—
	閉曲線PAINT	○	○	○	—/—
	直線	○	○	○	○
	スプライン	○	—	—	—
	マスク	○	—	○	—
	ブラシ	○	○	○	○
	ブラシ編集	○	○	○	—
編集	ルーベ(×2)	○	○	○	○
	ルーベ(×4)	○	○	○	○
	ルーベ(×8)	○	○	○	○
	ルーベ(×16)	○	○	—	○
	矩形COPY	○/○	○/—	○/○	○
	/MOVE				
	閉曲線COPY	○/○	○/—	○/○	—/—
	/MOVE				
	矩形変形	○	△(画面回転のみ)	回転のみ	—
	拡大/縮小	○	△(全画面のみ)	○	○
	上下反転	○	△(全画面のみ)	—(回転で可)	○
	左右反転	○	△(全画面のみ)	—	○
	シフト	—	△(全画面のみ)	—	○
	カラーチェンジ	○	○	○	—
	パレット編集	○	○	○	○
	モザイク	○	—	—	—
	ぼかし	○	—	—	—
文字	16ドット	○	○	—	—
	24ドット	○	○	○	—
	アウトライン	○	—	—	—
	斜体	○	○	—	—
	グラデーション	○	—	—	—
	影	○	○	—	—
	縁取り	○	○	—	—
外部入力	COLOR IMAGE UNIT	○	○	○	—
	IMAGE SCANNER	○	○	○	—
座標表示		○	○	○	○
方眼紙		○	○	—	○
アンドウ		○(要2MB)	△	○	—
画面数		1	3	1	2
スプライト・セーブ		—	○	—	○
アニメーション		—	—	—	○
子プロセス		—	○	—	—
オートデモ		—	○	○	—
おまけ		ZIMLOAD他	BASIC関数	ZIM↔GL3変換	

ギザギザのないグラフィック関数

アンチエリラスとは？

Tan Akihiko **丹 明彦**

というわけで、2次元グラフィックである。これまでは3次元グラフィックが主だったの、次元がひとつ落ちたことになるのだが、それはつまり、質的にも一段と落ちたことなのか？ いやいやとんでもない。2次元のほうが3次元よりもずっと身近で表現しやすいのである。そして表現しやすいぶん、人は精魂こめて絵を作り上げるし、質的にも高いものができる。そのことはOh!X誌に毎日のように送られてくるイラストを見てもわかる。とにかく、層の厚さが違うぶん、競争も激しいし、いいものしか残らない。これはとてもいいことである。

さて今回の目標は

これから紹介するのは、コンピュータのスクリーン上によりよい2次元の1枚絵を作るための道具である。といってもX-BASICのグラフィック関数とやっていることは基本的には同じ。1つひとつの関数の動作は非常にプリミティブなもので、現段階では「ペンと紙とスクリーンをキーボードに持ち換えた」と同じような感覚で使うことは、残念ながらまず無理である。優れたグラフィックツールであるZ'sSTAFFでさえ、ただペンをマウスに持ち換えただけなのとは少し違うのだが、それとは次元が違う。

今回制作しようというX-BASICの外部関数は、マウスから入力するといったユーザーインタフェースについては無視である。つまりその部分はユーザーであるあなたにお任せ、ということになる。用意したのはやや強力なラインやペイントなのだから、それをあなたがどう活用しようとまったく自由である。

X68000でラインやペイントを使った2次元グラフィックで良質なイラストを作ろうというのが今回の試みだといったが、こういう反論もあるだろう。「X-BASICにだってラインやペイントはあるぞ、どうしていまさら作り直す必要がある？」と。そ

の考えは甘い。X68000の標準グラフィック関数は、せっかくの65536色を生かしきっていないのである。

コンピュータで描いたイラストの多くがどうして雑誌の表紙を飾りうるだけのクオリティを持ちえないのか。よくできてはいるけどどこか違和感のあるイラスト。そのひとつの解答がここにある。輪郭に出てくる見苦しい階段、すなわち「ジャギー」である。

'90年のトレンドはドッター

その昔、人間デジタイザと呼ばれる人々がいて、変な奴と思われながらも尊敬を集めていた。かれらの道具はラインとペイントであった。当時はマウスなどという便利な道具は庶民の持つべからざるものであった。Z'sSTAFFのような操作が簡単なうえに強力な表現力を持ったグラフィックツールに至っては、夢のまた夢であった。

そこで彼らは方眼紙に下絵を描き、座標値を丹念に取りながらぼちぼちとキーボードから打ち込んでいたのであった。そしてラインで線を引き、中をペイントで塗りつぶす。

いまでも総天然色（ちと古いか）は常識でも、8色が主流であった時代のこと、微妙な筆づかいなどは表現しえようはずもない。そのため古来の名画を模写するような試みはあまりなく、彼らの興味はもっぱらアニメ絵に向いていた。パソコン使いとアニメファンの深い関係はこうしてできあがったのであろうか。

そして時代はアナログに向かい、高品質の絵を誰でも作れる、そんな期待を感じさせるマシンの登場を見た。X68000である。ところがその期待はまだ期待の域を出てはいないのかもしれない。

X-BASICでline()関数を使ってみた方は、およそ滑らかさが無いのに驚かれたのではないかと思う。もちろん、従来機種ではそれが当たり前のことだったのだが、せ

コンピュータグラフィックでの強敵のひとつジャギー。今回はこれを追放すべく、新しいグラフィック関数を揃えてみました。もっとエレガントなラインルーチンと高機能なペイントルーチンなどによる高画質な2Dグラフィックワールドを構築していきます。

めて65536色モードのときくらい、もっと目に優しい線がほしい……というわけでZ'sSTAFFに期待がかかるわけであるが、こちらでも残念ながら完全なサポートはなかった。

この件の解決法はいくつかある。

- ・中心部が濃く周りが薄いペンを指定して、ふつうに線を描く
- ・ただの線を描いてあとからぼかす
- ・あきらめる

3番目は問題外として、どれも自然で滑らかな線にはならない。さらに共通の欠点もある。これらの方法でそれらしく見えるように線を描けたとしよう。すると、こういう線で囲んだ内側をペイントしようとしてずっこけることになるのである。ペイントできないのである。いや、できることはできるがきちんと隅々まで塗りつぶしてくれないのである（手元にZ'sSTAFFのある方はお試しいただきたい）。というわけで最後の手段として、

- ・ルーペで拡大して1点1点描く
- ということになるのである。

現在あちこちで（市販ゲームのビジュアルシーンなどで）見かける比較的良質な画像のほとんどは、こうやって描かれている。現在のデジタルペインティング界を支えているのは、このドッターたちなのだ。

僕はこのルーペでドット打ちという作業を自分ではしたことがないので、はなはだ無責任な意見ではあるのだが、どう見ても非人間的な作業としか思えない。この点、人間デジタイザと似通っている。

しかし描いている本人は決してそうは思っていないであろう。この手の作業は慣れると苦しくはなくなるものである。それに つれて質も上がってくる。しかしどうしても職人芸になりがちである。いきおい選ばれし者の技術になってしまう。そして一般ユーザーからは変な奴とか閑人とかのレッテルを貼られてしまうのである。合掌。

今回はそこまでの質を追求するつもりはない。BASICから手軽に使えればよい。い

ろいろと遊べたらなおよい。そんな気持ちで作ってみた。

アンチエイリアシング

で、さきほどもちろっと出てきた「ジャギー」である。これは昔から再三いっているように、有限個しかないグラフィック画面のドットで、無限とでもいい細かさの画像を表現しようという要求のなかで、起こりうるべくして起こる問題である。サンプリング理論の言葉で「エイリアシング」という。

これを防ぐためには、視力の限界を超えた高い解像度のCRTを使うのが完璧な解決法であろう。しかしそんなものはないし、あっても化け物のように高価であろう。

ではどうするのか。うまいことフィルタをかけて、不連続に変化しそうなピクセルの輝度の変化を補間するというのが現在もっとも効果を上げている方法である。

黒い線を引いたつもりでも、その緑の部分には微妙に灰色のピクセルが並んでいて、遠くから見れば滑らかな線に見えるのである。境界をぼかしてごまかしているのと混同されがちだが、これはぼかし処理とはまったく異なるもので、アンチエイリアシングと呼ばれる。

百聞は一見にしかずというわけで、まずはなにもいわずにリスト1を実行していただきたい。いうまでもなくX-BASICのリストである。

図1 点列のデータ構造

線が4本画面に見える。そのうちいちばん上といちばん下の線は画面の中央でがたとんと1ドット上がり、2本の線を1ドットずらして継ぎ合わせたような印象である。

1本の線には見えない。

対して内側の2本はわりと綺麗な線に見える。そしてこの4本は、全体としては平行線である。右端と左端を見ると、確かに等間隔である。とすると、内側の2本はまっすぐな線に見えるようだが、状況から見て、どこかで1ドット上がっていかなくては辻褄があわない。

ここでZ'sSTAFFをお持ちの方は、ループを使って、内側の2本がどのような色使いをしているのかを見れば、アンチエイリアシングの原理がおぼろげにでもわかることであろう。

しかしこれではあまりにも応用がきかない。今回作ったのは、もっといろいろな線にも使えるようなプログラムである。その具体的なアルゴリズムの説明はあと回ししよう。とりえず使えることが大切だ。

使いかた

どういう形式で実現するか迷ったのだが、手軽に使えるX-BASIC外部関数という線に落ち着いた。その関数本体はC言語で書いている。

X-BASICの外部関数をCで書くときの注意や、コンパイルの手順などは囲みにしてあるのでそちらをご覧くださいとして、

いま、あなたの手元には今回作った外部関数anti.fncを組み込んだX-BASICが起動しているものと思って話を続けることにする。

関数のリファレンスマニュアルを表1に掲げる。anti.fncにはこの表にない関数も収録してあるが、隠し関数のようなものなので、とりあえずは表1に載っているものだけを使っていたきたい。

anti.fncを使いこなすには、表1や図1にも出てきている「点列」というデータ構造の把握が不可欠である。というよりもそれがほとんどすべてである（点列にはCのソースファイル中ではPTSという型を与えている。pointsを略して命名した）。

点列の基本単位は整数3個で、それが（頂点の数+1）個並ぶかたちになる。X-BASIC上では、

```
int pts(10,2)
```

と宣言する。BASICの配列の宣言は、Cのそれと少し違っていて、同じ宣言をCでは、

```
int pts[11][3];
```

```
10 /* アンチエイリアシングの原理
20 /* 1ドットの段差を持つ線
30 screen 1,3,1,1
40 /* アンチエイリアシング
50 for x=0 to 511
60   i=32*x/512
70   pset( x,199,rgb(i,i,i) )
80   pset( x,200,rgb(31-i,31-i,31-i) )
90   pset( x,204,rgb(i,i,i) )
100  pset( x,205,rgb(31-i,31-i,31-i) )
110 next
120 /* ノンアンチエイリアシング
130 line( 0,210,511,209,65534 )
140 line( 0,195,511,194,65534 )
```

dim int pts(N, 2)で定義する。Nは点列のおおよそのサイズ。

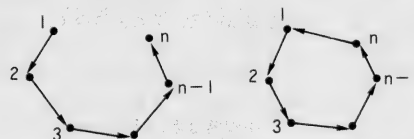
	0	1	2	
0	n	0/1	0/8	ヘッダ
1	x ₁	y ₁	w ₁	頂点の情報
2	x ₂	y ₂	w ₂	
3	x ₃	y ₃	w ₃	
⋮	⋮	⋮	⋮	
n-1	x _{n-1}	y _{n-1}	w _{n-1}	
n	x _n	y _n	w _n	未使用
n+1				
⋮	⋮	⋮	⋮	
N				

pts(i, j)の意味

〔ヘッダについて〕 (i=0)

pts(0, 0) = n 頂点数(≤N)

pts(0, 1) = 0 または 1 点列のタイプ



pts(0, 1) = 0

多角形は開いている

pts(0, 1) = 1

多角形は閉じている

pts(0, 2) = 0 または 8

オーバーサンプリング座標のフラグ

{ pts(0, 2) = 0

オーバーサンプリング処理がされていない

pts(0, 2) = 1

オーバーサンプリング処理済み

〔頂点について〕 (1 ≤ i ≤ n)

(x_i, y_i) 第 i 点の座標

w_i 第 i 点での線の太さ(aa_lines()関数)

とする。BASICは添え字の最大値を、Cは1次元あたりの要素の数を基本にしているからだが、あとの参照や代入のしかたは両者ではほとんど変わらない。

点列の宣言は前述のとおり2次元配列で行うが、第1添え字(10)は頂点の数の最大値というか、その目安を適当に決めて設定する。たとえば複雑な形なら値を大きくする。曲線を記録する(後述)ときも大きくする。第2添え字のほうは2に固定である。

点列の構造について少し解説しよう(図1)。頭から3要素、pts(0,j)は少し特殊

で、ヘッダと呼んでいる。pts(0,0)には実際の頂点数が、pts(0,1)には点列のタイプが、pts(0,2)にはオーバーサンプリング倍数が入る。

点列のタイプは2つに分かれる。それを理解する助けとして、1本の紐を想像してもらいたい。その紐が点列を表している。いま、その紐の端と端を結んだとする。その状態が、点列タイプ=1の状態、循環していると名づける。要するに閉じているわけである。そうでない、開いている状態が点列タイプ=0というわけである。

今回のプログラムの作り方

X-BASICの外部関数をC言語で作るわけであるが、今回のプログラムは、

- ・内蔵の関数(機能)が比較的多い
- ・それぞれの処理が多少複雑
- ・したがってプログラムサイズが大きい
- ・たったひとつの関数をデバッグするのにいちいち全部コンパイルしなおしてはやりきれない

というわけで、分割民営化、じゃない、

分割コンパイル

の採用に踏み切った。複数のソースファイルを別々にコンパイルして、最後にリンクを使ってひとつにまとめるやり方のことである。僕も今回ほどバラバラにしたのは初めてだが、いざやってみると非常に快適である。

0) 環境

最初に開発環境を確認しておこう。

使うCコンパイラはXCがGCC。コンパイラはどこに置いておいてもいいが、パスは通っていないてはならない。コンパイラのほかにアセンブラ(as.x)とリンクラ(lk.x)が必要である。これらにもパスを通しておくこと。当然ながらテキストエディタも必要。僕はmicroEmacsを使っているが、標準的なのはed.xであろう。

設定しておかなくてはならない環境変数もいくつかある。autoexec.batなどに次の設定がされているかどうか確認しておくこと。システムがAドライブでRAMディスクがFドライブの場合、

TEMP F:

SET lib=A:¥LIB

SET include=A:¥INCLUDE

BASICの入っているディレクトリは、

A:BASIC2¥

とする。そうでない方は各自のシステムにあわせて読み換えてほしい。

ほかに大切なのはインクルードファイル(*.h)およびCのライブラリ(*lib.a)である。それぞれ、

A:¥INCLUDE¥

A:¥LIB¥

に収めておくこと。C compiler PRO-68Kのシステムディスクの設定なら基本的には安心してよい。そうそう、GCCの場合は、gnulib.aというライブラリもあるが、これもA:¥LIB¥に収めておけばよい。

1) ソースリスト作成

環境設定ができたなら、さっそくソースリストを作ろう。打ち込むリストは次のとおり。すべてふつうにテキストエディタで打ち込む。

・anti.s (外部関数ヘッダ)

・anti.h (マクロ定義ファイル)
・main.c (引数リスト宣言)
・pts_curve.c (自由曲線の発生)
・pts_procs.c (輪郭の処理)
・aa_lines.c (輪郭描画)
・aa_scanconv.c (多角形塗りつぶし)
・aa_paint.c (閉領域ペイント)
・aa_procs.c (タイル・トーン処理)

一度に全部打ち込むのもおっくうなので、テストしながら作業を進めたい方や、必要ない関数を打ち込みたくない方は、そういう関数の名前だけ書いて中身を書かない(return(0))だけは入れておいたほうが安全だが)という手が使えるので参考にいただきたい。

2) コンパイルおよびアセンブル

ソースリストを打ち込んだら、それぞれをコンパイルする。ただし、anti.sとanti.hは例外。anti.sはアセンブラ(as.x)でアセンブルする。

as /u anti.s

エラーがなければ、anti.oというファイルができる。anti.hのほうはただのインクルードファイル(それぞれのCのソースにインクルードして使う)で、それ自身を単独でコンパイルする必要はない(してもなにもできてこない)。

さて、Cのプログラムのコンパイルだが、こちらもふつうどおりではない。分割コンパイルなので、リンクフェイズまで一気に突っ走ってはいけない。~.oの段階で止め、最後にリンクするのが分割コンパイルである。だからリンクフェイズの直前でコンパイルをやめるスイッチをコンパイラに与えてやらなくてはならない。これがXCとGCCでは違って、それぞれ、

cc /L ~.c

gcc -c ~.c

である。また、GCCの場合は最適化オプションが豊富なので、それもついでに与えよう。いちいち長たらしいオプションを打ち込むのは面倒なので、次のようなバッチファイルを作ることとする。これもテキストエディタで書く。ファイル名は仮にcompile.batとしよう。

(XCの場合)

cc /L %1.c

(GCCの場合)

gcc -c -O -fstrength-reduce -fomit-frame-pointer -finline-functions %1.c

バッチファイルができたなら、

compile main

compile pts_curve

:

compile aa_procs

オーバーサンプリングについては、もう少し後ろで説明するが、予備知識として簡単にしておく、今回の目玉であるアンチエイリアシングに使う技法である。

ひとつのピクセルをより細かく分割して図形を描き、出力する段階で平均すれば、最終的に出てくる図形の輪郭が滑らかになるという思想に基づいている。座標系を、ピクセルのサンプリング周波数よりもっと細かく取るから、オーバーサンプリングと呼んでいる。今回は8倍オーバーサンプリングとしたので、pts(0,2)には0か8が入る。

と各ソースごとにコマンドラインから実行する。もしエラーが発生したりバグを取ったりしたファイルがあれば、そのファイルだけをコンパイルしなおせばよい。

3) リンク

ここまで無事終了したら、~.oというファイルが8つできていることであろう。そこで仕上げるリンクフェイズ。

lk /o anti.fnc anti.o main.o pts_curve.o pts_procs.o aa_lines.o aa_scanconv.o aa_paint.o aa_procs.o %lib%¥clib.a (%lib%¥gnulib.a) %lib%baslib.a

カッコ内のgnulib.aというのはGCC専用のライブラリで、いうまでもなくXCでコンパイルする人には必要ない。/oオプションを使って、ふつうならanti.xとなる出力ファイルの名前を外部関数の名前anti.fncにする。実はX-BASICの外部関数の正体は実行形式ファイルと同じである。ただ名前がそうになっていないだけ。

4) インストール

あとはX-BASICにできたてのanti.fncを組み込むだけである。まずBASICのディレクトリにanti.fncを転送する。

copy anti.fnc A:¥BASIC2¥

それからBASICのディレクトリ上のコンフィギュレーションファイルをテキストエディタで書き換える。標準ではbasic.cnfというファイル名である(X-BASICは/cオプションを使って指定したコンフィギュレーションファイルで立ち上げることもできる)。

以下はその一例である。大切なのは最後の1行。

FREE = 128

WIDTH = 64

BEEP = ON

CAPS = OFF

FUNC = GRAPH

FUNC = PIC

FUNC = ANTI

ほかにも音楽関係の外部関数を組み込んでおけば、音楽を演奏しながら絵を描くという芸当もできるだろう(してなんになる)。ところで下から2行目のpic.fncというのは、やはり本誌6月号の付録ディスクについていたPIC形式の画像ファイルをセーブ/ロードする外部関数。描画の遅いanti.fncにとってはとてもありがたい相棒である。

これでやっと使えるところまでこぎつけた。正直いって、Cとアセンブラを扱いたない人にはこんな説明は退屈なだけかもしれない。

pts(i,j)は、 $1 \leq i \leq \text{pts}(0,0)$ である i については i 番目の頂点の情報を格納する。pts(i,0)にはx座標が、pts(i,1)にはy座標が、pts(i,2)には線の幅がそれぞれ入る。

それでは動作チェックも兼ねて簡単な使い方を練習しよう。まずは点列の宣言の方法から。

例1) V字型

```
dim int p1(3,2) = { 3,0,0
,100,100,8
,200,300,8
,300,100,8 }
```

例2) 三角形

```
dim int p2(3,2) = { 3,1,0
,100,100,8
,200,300,8
,300,100,8 }
```

この2つのサンプルのあいだでは、点列タイプ (pts(0,1)) だけが違うことに注意しよう。

点列の定義ができたら、それを使ってなにか描いてみよう。その前に、完全に制作者 (要するに僕) の都合なのであるが、点列をオーバーサンプリング座標系に変換しなくてはならない。変換をかけておかないと、この先出てくるほとんどの関数が使えない。ま、ここはおまじないとも思っておこう。

```
pts_oversample(p1)
```

```
pts_oversample(p2)
```

次に、画面モードを65536色モードに変える。ちょっと手抜きなことに、描画関数の中に画面モードのチェックを入れていないので、忘れずに実行しておくこと。

```
screen 1,3,1,1
```

では先ほど作った三角形を画面に出してみよう。

```
aa_lines(p2, rgb(31,31,31))
```

なかなかダルいが、おしまいまで待とう。白い三角形が出てくると思う。

お次はいまの三角形の頂点を通る曲線を作ってみよう。それにはまず、曲線を格納する配列をひとつ用意する。というのも、曲線は短い線分をたくさんつなげてそれらしく見せるようにしているからだ。そのため、ある程度多くの頂点も記録できるように大きな配列を用意する。余裕を持って、

```
dim int p3(1000,2)
```

と大きめに宣言しておき、すかさず、

```
pts_curve(p2,10,10,p3)
```

を実行。

pts_curve()は曲線を生成するだけの関数なので、画面にはなにも出ないはずだ。ちょっとした戻ってくるので、できた曲

線を見てみよう。さっきと同様に、

```
aa_lines(p3,rgb(31,0,0))
```

今度は赤い色で三角形のカドを取ったような曲線が出てくるはず。

さてここでいったんご破算願おう。

```
wipe()
```

そして新しい気持ちでもっと妙な形を試してみることにする。

```
dim int p4(6,2) = { 6,1,0
```

```
,100,100,8
```

```
,200,300,8
```

```
,300,100,8
```

```
,400,400,8
```

```
,300,200,8
```

```
,200,400,8 }
```

例によってオーバーサンプリング座標に変換するおまじない。

```
pts_oversample(p4)
```

この「N」をひっくり返したような多角形の頂点を通る曲線を作る。

```
dim int p5(2000,2)
```

```
pts_curve(p4,8,8,p5)
```

さっきは輪郭線だけだったが、今度はこの曲線の内側も塗りつぶしながら描く。

```
aa_scanconv(p5,0,65534,0,0)
```

白い変な形が現れる。その中を赤でペイントしてみよう。

```
aa_paint(250,250,0,rgb(31,0,0),0,0)
```

ちなみに、このaa_paint()の代わりに、paint(250,250,rgb(31,0,0))

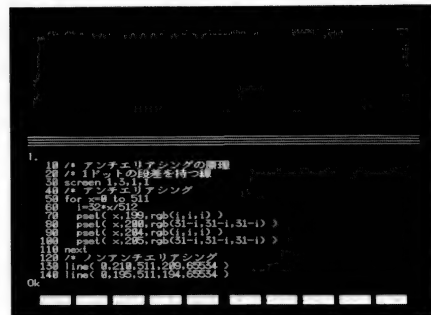
を実行してみると、aa_paint()がアンチエイリアシングに対応しているありがたいペイント関数であることがわかることだろう。

以上の動作に支障がなければ、ほぼバグはないと考えていいだろう。表1の関数リファレンスを参照しながら、上の例題の数値をあちこちいじって実行してみよう。そして、それぞれの関数がどういう機能を持ち、どんなパラメータを与えるとどんな動作をするか、そういうことを理解して、さらに難しい作品へと進んでいってほしい。

アルゴリズム解説

ソースリストが思ったより大きくなってしまい、我ながら驚いている。こんなものの説明をすることを考えるだけで胸焼けである。ま、すべてはソースリストが語ってくれるということで、コーディングするうえでの細かい注意は、ソースリストに入れたコメントに頼ることにし、ここではアルゴリズムの心を語ることにする。

今回の外部関数を構成するための主要なアルゴリズムはいくつかある。幸いなこと



アンチエイリアシングの奇跡

に、過去のOh!X誌ですでに僕が紹介しているものも多いので、適宜参照していただきたい。

オーバーサンプリング

アンチエイリアシング技法のなかでもっともポピュラーな方法のひとつが、このオーバーサンプリングである。レイトレーシングやZバッファといった3次元CG技術をアンチエイリアシング対応にする場合、必ずといっていいほど用いられるのもオーバーサンプリング。

ここまでの説明でもちらっと触れているのだが、まず事実として、ピクセルのサンプリングレート (要するに解像度) はかなり高いように見えて、人間の目をごまかしおおせるほどには高くないということがある。そこで多色表示の利点を生かすことが考えられた。ともすれば急激で不連続的になりがちなピクセルの輝度変化をもっと滑らかにし、曖昧な (少し語弊があるが) 輪郭を作れば、目に優しい画像ができあがる。

そのために、いったんピクセルよりも高いサンプリングレートで画像を生成しておく。このときの最小の処理単位は、ピクセルよりもさらに小さな画素であり、サブピクセルと呼ばれる。

ちなみに1本のスキャンラインも数本のさらに細いスキャンラインに細分されることになり、サブスキャンラインと呼ばれる。今回の外部関数では8倍オーバーサンプリングを採用している。この場合1ピクセルは $8 \times 8 = 64$ サブピクセルからなる。

描画アルゴリズムは従来の (オーバーサンプリングを用いない) アルゴリズムを拡張して使う。ただ処理単位がピクセルでなくサブピクセルになっているだけである。

そして、1ピクセル中の全サブピクセルの輝度を平均してスクリーンに出力すれば、粗いピクセルにそれ以上の解像度を持たせたのと同等の効果が得られるという仕掛け

になっている。

誤解を恐れずにいうなら、アンチエイリアシングは人間の目を巧みにごまかす技法であるともいえる。もちろん、ピクセルをよく見ればそんなごまかしはすぐわかってしまうし、1ピクセルを下回るような細かい図形には効果が薄くなってしまおうといった欠点はあるものの、いたずらに解像度を上げるよりもずっといい方法なのである。

今回の描画アルゴリズムでは、サブピクセルの輝度を1つひとつ配列に持っておくことはしなかった。2次元なので、基本的に隠面処理など考える必要はないし(*), それならば「いま描画しようとしている図形が各ピクセルのうちいくつのサブピクセルを占めているか」という情報だけが重要だとわかる。これをピクセルあたりの寄与率と呼ぼう。以後は α という記号を使うことにする。

8倍オーバーサンプリングの場合、サブピクセル数は0から64の値をとる。 α はこれを64で割った値、つまり $0 \leq \alpha \leq 1$ の間の値をとる。ピクセルと図形がまったく重ならない場合は $\alpha = 0$ だし、ピクセルを図形が全部覆っている場合は $\alpha = 1$ 。境界部だけで α は0でも1でもないいろいろな値をとる。

α は一般に実数だが、プログラム上は実数よりも整数のほうが取り扱いが楽なので、ひとまず $0 \leq \alpha' \leq 64$ で格納しておき、最後に64で割っている。これでも結局は同じである。

スクリーン出力の段階では、 α 合成と呼ぶ方法を用いる。背景が真っ黒な場合は α がそのまま輝度になるのだろうが、もちろんいつてもそんなことがあるはずはなく、ふつうは、適当な比で図形の色と背景の色

を合成しないと、輪郭が変になってしまう。この比に α を用いるのである。つまり次の比で混合する。

図形の色 : 背景の色 = $\alpha : 1 - \alpha$

参考) この方法の画質をもっと上げる方法として、重みづけ平均化をすることも考えられる。 α を出す段階で、ピクセルの中心部のサブピクセルほど α に大きく寄与するようにプログラムを組んでおくのである。今回採ったのは単純平均化で、どのサブピクセルも同じ重みをもっていることになっている。

Bresenhamのアルゴリズム

昨年解説したZバッファアルゴリズムの前フリとして線分描画を説明した(1989年7月号)。一般に線分の傾きは実数である。実数である線分の傾きを相手にしながらも、Bresenhamアルゴリズムはすべての演算を整数ですませてしまう。このアルゴリズムは、実に応用が広い。たとえば本誌5月号のグラフィック拡大縮小にも使っている。

Bresenhamアルゴリズムの核となる部分を以下に示す。 $(x1, y1)$ から $(x2, y2)$ へ色 c で線分を引く。ただしここでは $x1 < x2, y1 < y2$ である。ほかの場合についてもそれほど難しくない拡張で対応できる。

Bresenhamアルゴリズムの基本的な考え方は、ピクセルの中心と真の線分との上下関係を比べ、真の線分にもっとも近いピクセルを点灯していくというだけのことである。この上下関係を比べるのに、誤差と呼ぶ量 e を使って処理を効率的にしている。

$dx = (x2 - x1);$
 $dy = (y2 - y1);$
 $e = -dy;$ (誤差の初期値)

```
for (x=x1, y=y1; x<=x2; x++) {
    pset (x, y, c); (ピクセル点灯)
    e += (2*dy);
    (1ピクセルあたりの真の線分の上昇分)
    while (e>=0) {
        (真の線分が上にあるあいだは)
        y++;
        (ピクセルの座標を上げる)
        e -= (2*dx);
        (その分だけ真の線分との距離を詰める)
    }
}
```

もっと詳しく知りたい方は1989年7月号の記事を参照してほしい。

ライン

ただの線分ならば上のBresenhamアルゴリズムを使うのだが、アンチエイリアシング対応となるとそう簡単にはいかない。しかも今回は欲張って、線分の幅を変えられるようにしたのでよい厄介である。

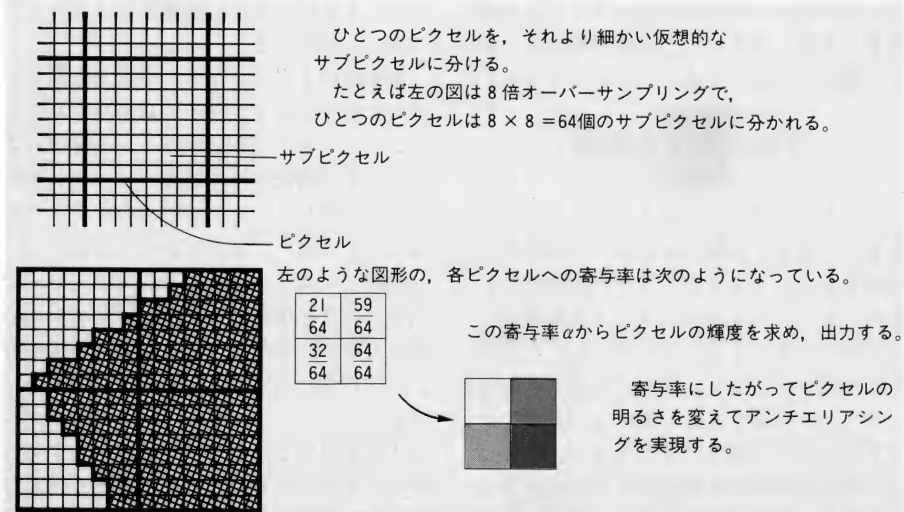
それでは、 $(x1, y1)$ から $(x2, y2)$ へ幅 w で線分を引くことを考えよう。といってもそれほど難しいことではない。まず描きたい太い線分を1ピクセル間隔で切る。イメージとしては輪切りである。そしてそのひとつひとつ(幅1ピクセルで長さ h ピクセルの小線分)をスクリーンに張り付けるのである。切り口の長さ h は、ピタゴラスの定理(おお懐かしい)を使って求めることができる。

ここまでわかればあとは簡単。まず太い線分の下端(これも線分になる)を通常の線分と同じようにBresenhamアルゴリズムで発生させる。

具体的には $(x1, y1 - h/2)$ と $(x2, y2 - h/2)$ を結ぶ線分、すなわち太い線分の中心から h の半分だけ下にずれた線分である。そして、この下端の線分の上に長さ h の小線分を並べていけばいい。これは、まっとうに描けば傾いた長方形になるはずの太い線分を、平行四辺形で近似したことになる。あまり線分が太くないうちはたいして不都合はおきないが、太くなってくると不自然さも目立つし、ときには破綻することもある。

(*) 3次元CGだとさすがにこんないい加減なことではすまされず、きちんとサブピクセル数だけのZバッファなりを用意し、隠面処理をきちんと終えてから合成するという手順が要求される(これはあくまで原理的な話で、実現するうえではもっと効率的な方法も提案されている)。

図2 オーバーサンプリング



る。このことはあとで触れる。

いずれにせよこれで太い線分は描ける。あとはこれをオーバーサンプリング座標系で処理し、寄与率 α をピクセルごとに求めてから α 合成を行うように拡張するとよい。ここから先は単なる力仕事である。また、`aa_lines()`関数はただ1本の線分ではなく数個~数百個の点列を結んで連続描画を行うので、それ相応の処理も考える必要がある。

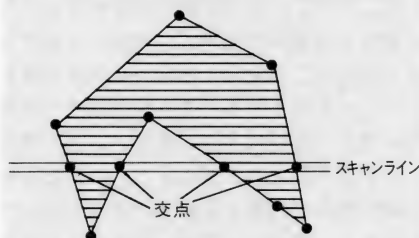
特にひとつの線分から次の線分に移るときは、前者の終点での寄与率を記憶しておいて後者の始点へとつなげていかなければならない。線分1本1本ごとに寄与率を初期化していたのでは、線分の継ぎ目継ぎ目でピクセルが暗くなってしまうからである(これは現実に失敗した)。

さきほどのめかしておいた欠点を説明しよう。`aa_lines()`関数では、傾きが小さいときはx方向に、傾きが大きいときはy方向に処理するようにループを組んでいる。また、太い線分といっても前述のとおりに平行四辺形で近似しているだけである。

そこで次のような事態は当然予想される。幅の太い曲線を描く場合を考えよう。その傾きは最初大きくてだんだん小さくなっていく。最初はy方向で処理していたのが、ある1点を境にx方向で処理するようになる。ここで曲線は、実にみつともないことに、まるでぽきんと折れたように欠けてしまうのだ。残念ながらこれを解消するうまい方法が考え出せなかった(下手な方法なら考えられないこともない)のでそのままにしてある。で、たいへん申し訳ないが、対抗策として、

- ・あまり太い線分は描かせない
- ・太い線分を描かせる場合は、傾きをうまくコントロールして曲線が折れないように工夫する
- ・どうしても自由な傾きで太い曲線を描きたいのであれば、面倒でも「太い曲線の輪

図4 ソリッドスキャンコンバージョン



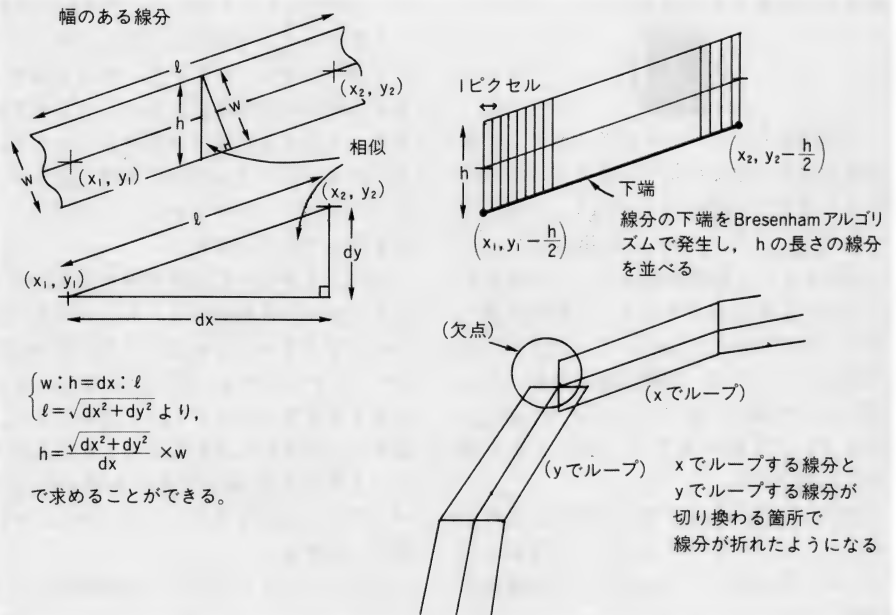
ソリッド領域をスキャンライン単位に細分する。スキャンラインと輪郭の交点は、Bresenhamアルゴリズムで求める。

郭」を作り、次のスキャンコンバージョン `aa_scanconv()` 関数で描かせる。スキャンコンバージョンのほうはどんな曲線に対しても破綻することはないなどとしていただきたい。

ソリッドスキャンコンバージョン

多角形を描画するもうひとつの方法で、上の`aa_lines()`がワイヤーフレームモデルだとしたら、こちらの`aa_scanconv()`はサーフェスモデルだといえるし、2次元ソリッドモデルだともいえる。要す

図3 線分の描画



リスト2

```
10 screen 1,3,1,1
20 fill( 0,0,47,47,rgb(0,0,31) )
30 symbol( 1,1,"色即",1,1,2,rgb(31,0,0),0 )
40 symbol( 1,25,"是空",1,1,2,rgb(31,0,0),0 )
50 symbol( 0,0,"色即",1,1,2,rgb(28,28,0),0 )
60 symbol( 0,24,"是空",1,1,2,rgb(28,28,0),0 )
70 tile_get( 0,0,0,47,47 )
80 tone_get( 0,0,0,47,47 )
90 fill( 0,0,47,47,rgb(16,16,16) )
100 tone_get( 1,0,0,47,47 )
110 wipe()
120 dim int p(10,2)={3,1,0
130     ,128,128,0
140     ,256,384,0
150     ,384,256,0}
160 pts_oversample( p )
170 dim int p1(10,2)={7,1,0
180     ,64,128,0
190     ,128,384,0
200     ,192,128,0
210     ,256,384,0
220     ,320,128,0
230     ,384,384,0
240     ,448,128,0}
241 dim int p2(2000,2)
242 pts_oversample( p1 )
243 pts_curve( p1,8,32,p2 )
244 whitepaper()
245 aa_lines( p2,0 )
250 /*aa_scanconv( p,1,0,0,0)
260 /*whitepaper(): aa_scanconv( p,0,rgb(31,0,0),1,0 )
270 aa_scanconv( p,1,0,1,1 )
```


寄ろう。まず多角形を細分する作業は、多角形とスキャンラインの交点の座標を求める処理に相当するが、これは輪郭をBresenhamアルゴリズムで発生すれば容易に求めることができる。またスクリーンに張り付ける作業は、求めた交点の間に線分を引く処理に相当する。やはり詳しい話は1989年7月号に譲る。

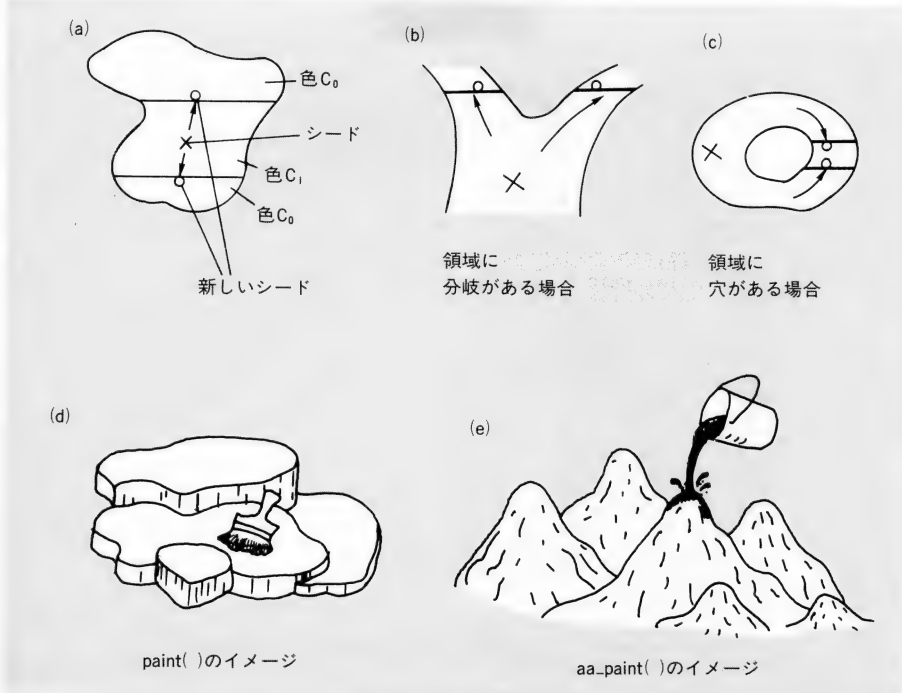
アンチエイリアシング化に際しては、先ほどのaa_lines()と同様のことをする。まずすべてオーバーサンプリング座標系で計算する。もちろんスキャンラインではなく、サブスキャンライン単位で処理するのである。そしてピクセルごとに寄与率 α を求めて背景と α 合成を行う。

ペイント

今回構成したペイントのアルゴリズムは実は本邦初ではないかと自負している（もちろん井の中の蛙かもしれない）。使い方の説明のところで、anti.fnc組み込みのaa_paint()と標準のpaint()の両方でペイントを実行してもらって、結果の違いを見ていただいた。paint()のほうでは、内側は赤いものの、曲線の縁の部分に白っぽいゴミが残り、変だった。対してaa_paint()だと隅々まできっちりと赤く塗ることができる。

ペイントのアルゴリズムをご存じだろうか。コンピュータグラフィックではシードフィルと呼ばれることもある。まず出発点

図5 ペイント



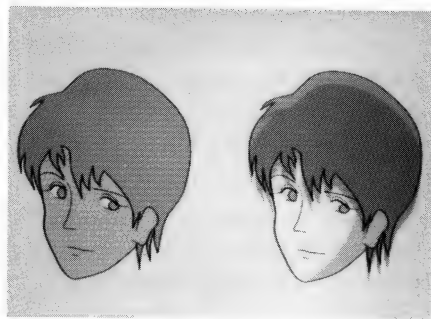
がある。これをシード（種）と呼ぶ。シードとなったピクセルの色 c_0 を記憶しておく。あとは、シードと同じ色をしていて、なおかつシードから到達可能なピクセルをすべてピックアップして、目的の色 c_1 で塗りつぶす。

「到達可能なピクセルを探す」アルゴリズムで一般的なのはFIFOバッファを使うアルゴリズムである。FIFOはファイフォと読み、先入れ先出し(First In First Out)方式でデータを格納する倉庫のようなものである。待ち行列といったり、キュー(queue)といったりする。ついでにスタックは後入れ先出し(Last In First Out)の倉庫で、LIFO(ライフォ)バッファといえる。

まずシードから左右を（同じスキャンライン内で）サーチしていく。サーチはピクセルの色が c_0 以外の色になったところで止める。左右ともにサーチが終わったとき、そのスキャンラインの中で到達可能なピクセルがピックアップされたことになる。それを色 c_1 で塗りつぶす。

次に、上で作った到達可能な区間の上下のピクセルの色を調べる。もしその中に色が c_0 のピクセルがあれば、その座標をFIFOバッファに入れる。色が c_0 の領域は1本のスキャンラインにひとつとは限らない。途中でふたまたに分かれることもあるだろう。そうした領域の代表点を過不足なくバッファに入れるように、コーディングの際は工夫する。

ひとつのシードについての処理がひとと



タイトルも使える

おり終わったら、FIFOバッファから座標を1組取り出してきて、新しいシードにする。もしそのシードがすでに色 c_1 で塗りつぶしてあった場合は、そのシードを捨てる。そんなことが起こるのかと不思議に思う方もいらっしゃるだろうが、たとえばドーナツのように穴のあいた領域をペイントするときは、ぐるっと回ってきた色 c_1 の領域がぶつかるので、FIFOバッファに入れたときは色が c_0 でも、FIFOバッファから出すときは色が変わっているということも起こりうるのだ。衝突したときに、どちらかのシードが無効になるわけだ。

さて、シードが有効なときは、そのシードから出発して上と同じことを繰り返す。そしてFIFOバッファが空になったとき、ペイントも終わる。

以上はふつうのペイントのアルゴリズム。だがオリジナルのペイントルーチンでも、基本は同じである。やはり到達可能なピクセルをピックアップし、シードを更新しつつ色を変えていけばよい。違うのは、「到達可能」を判定する条件である。ふつうのペイントでは、シードの色と同じであることがその条件であった。しかしここではアンチエイリアシングをかけた領域には対応できない。なぜなら、アンチエイリアシングは縁の色を少し暗くすることで滑らかに見せているものだから、当然縁の部分はシードと色が同じはずはないのである。したがって縁まできっちりと塗ることは不可能。

そこで新しい条件を作る必要がある。その条件とは極めて簡単で、現在いるピクセルと同じか、またはより暗いピクセルをたどっていくのである。こうすればサーチは暗いほうへ暗いほうへと進んでいき、明るくなりそうところで止まる。これなら黒い線で囲んだ内側なら確実に隅まで塗ってくれるし、隣の白い部分にはみ出すこともない。

イメージとしては、山の頂上からペンキを流す図を想像していただきたい。ペンキは下へ下へと流れ、一番低いところで止ま

る。ほかの山を上っていくようなことはしない。

注意をうながしておくが、頂上、つまり白い部分の一番明るい点にシードを置かないと、やはり正確に塗ってはいくれない。

以上からもわかるとおり、オリジナルのペイントアルゴリズムでは、オーバーサンプリング座標は使わない。かわりに、ピックアップした点の輝度を寄与率 α のように考えて（というよりもアンチエイリアシング描画におけるピクセルの輝度はもともと α を反映したものなのだが）、 α 合成に似たことを行う。単純に色c1で塗るのではなく、

c1に α をかけた色で塗るのだ。

したがって、ペイントずみの領域を判定するには、色がc1であるかどうか、という見分け方が使えない。ペイントずみの領域には少し暗いc1というものもある。そこで今まで出番のなかった輝度ビットをフラグとして使うことにした。X68000のカラーコードは16ビットで、上から5ビットずつ緑、赤、青の3原色が割り当てられる。そして最下位の1ビットが輝度ビットなのである。RGBと独立になっているので妙な用途に使われることが多い。Z'sSTAF Fでもマスキングに用いている。

CとBASICの相性

X68000以前は、BASICの機能拡張といえば、メモリの空きエリアを捜して処理ルーチンを組み込んだり、パッチを当てたりといった、どことなく超絶技巧の香りが漂う技術であった。X-BASICでは、機能拡張を正式に許し、その仕様を公開している。さらに書くことと思えばCで書いたっていいのである。この姿勢には頭の下がる思いである。と同時にプログラマが甘やかされそうな気がしなくもない。

さて、そのX-BASICの外部関数はCで書くことができるのだが、いくつかの制限がある。

Cで素直に書けない部分について

いきなり矛盾したことをいっているようだが、BASICインタプリタと外部関数のインタフェースを取る段階で、どうしても純粋なCだけでは無理な部分があるのである（しかし素直でなくなりさえすれば簡単に書ける。ここがCの頼もしさであり、同時に怖さでもある）。具体的には、

- ・外部関数のヘッダ
 - ・引数を渡す
 - ・戻り値を返す
 - ・外部関数エラーのコードを返す
 - ・エラーメッセージのアドレスを返す
- 部分である。

このうち戻り値に関しては、今回作った関数はみんなvoid型ということにしてしまったので問題は起きない。

それにもかかわらず、Cのソースリストでの関数の戻り値がint型（typedefを使ってFUNC型としてはあるが）なのは、Cの関数の戻り値を実はBASIC側ではエラーコードとして受け取るためである。Cの関数は（整数型の）戻り値をd0レジスタに入れてリターンするというしきたり（?）があり、またBASICのエラーコードはd0で受け取るという規則になっている。return(0)で戻れば関数が無事に終わったことを、return(1)で戻ればなにかトラブルが起きたことをBASICに知らせることができる。エラーが起きたことがわかれば、インタプリタはエラーメッセージを出し、ビープ音とともにプログラムの実行を中断してくれる。結局どちらもCで書くことができるのでこれも問題ない。

エラーメッセージが問題である。a0レジスタにエラーメッセージの先頭アドレスを入れて返さなくてはならない。これはさすがにCで書くことはできない。しかしC言語には、インラインアセンブラといって、ソース中にアセンブラ

のコードを直接埋め込むという技が用意されている。まっとうなCコンパイラなら必ず使えるこの技は、当然X68000上のCコンパイラ、つまり標準のXCでも本誌6月号の付録ディスクで配布したGCCでも使える。ただ両コンパイラでのインラインアセンブラの使い方は少し違って、XCでは、

```
#asm
    lea __errmsg,a0
#endasm
であるが、GCCでは、
asm ("lea __errmsg,a0");
である。ここで、__errmsgはエラーメッセージを格納しているアドレスである。
```

GCCは本来の活動の舞台がUNIXなので、GCCの書き方はUNIX標準のCと同じである。C言語界全体を見渡せばむしろXCの作法がローカルな部類に入るのだろうが、そんなことはX68000でプログラムを作っている僕らにはなんの関係もない。どうにかして両者の違いを吸収する必要がある。

GCCのドライバはコンパイルに際して、プリプロセッサに、

```
#define __GCC__
と指定したのと同じことを自動的に行う。ま、環境変数みたいなものだ（本当は全然違う）。今回はこれを利用して条件コンパイル（#if～#else～#endif）をする方式を採用した。
#ifdef __GCC__
asm ("lea __errmsg,a0");
#else
asm
    lea __errmsg,a0
#endif
#endif
```

しかし読者の方はこんな面倒なことをする必要はない。各自の使いたいコンパイラにあわせて部分だけを打ち込めばそれでよい。

お断りしておくまでもないと思うが、GCCだけ手に入れてもコンパイルはできない。コンパイルに際してはアセンブラとリンクとXCのライブラリが必要なので、XCつまりC compiler PRO-68Kは必ず持っていないとてならない。

ちなみにエラーメッセージであるが、グローバル変数の文字列として宣言するのがコツである。関数の外側で、文字列（char型配列）へのポインタとして、たとえば、

```
unsigned char errmsg[] = "エラーだよ";
と宣言するとよい。こちらの変数名の頭にはア
```

また、 α の値は、aa_paint()では赤成分の輝度を代表で持ってくることにした。これによりどんな不都合が起こるかという、たとえば真っ青な領域は塗れないのである。赤成分がないので、全部黒と見なされるのだ。その他、明るいとこから暗いところへと塗るアルゴリズムのため、暗いところから出発して明るいほうに塗っていくような塗り方もできない。

以上のように妙な制限が多いので、白地に黒く線を描いてその中を塗るという使い方をすすめる。ついでにもうひとついっておくと、あまり細い領域を塗ろうとす

ンダーバー（_）がつかないことに注意。コンパイラは、ソース中のラベル（関数名や静的変数名）にアンダーバーをひとつつけてアセンブラに渡すが、すでに述べたとおり、インラインアセンブラの中身にはいっさい手を出さないで、こんな配慮が必要である。いくらCで書けるといっても、アセンブラの知識が少しはないと、外部関数は書けないのだ。

さて順番が前後してしまったが、外部関数ヘッダである。これはもう純粋にアセンブラで書かないとしようがない。もちろんインラインアセンブラは使えるが、上述の条件コンパイルと同じことを2回書く必要があるので、量が多いだけにユウウツである。ま、関数内で渡さなくてはならないエラーメッセージならともかく、ヘッダである。無理にCのソースリストの中に埋め込む必要もない。ヘッダは独立なファイルにした。それがantisである。

そしてもっともやっかいなのが引数の渡し方である。Cは引数を4バイトないしは8バイト単位でスタックに積み、関数に渡す。むしろ呼ばれた関数側でも4バイト、8バイト単位で受ける。ところがX-BASICは引数を10バイト単位でスタックに積み、外部関数に渡す。10という数字はCにとってはとても半端な数字である。おかげでBASICからもらってきた引数を、Cのほうでストレートに受け取ることができなくなってしまっている。

で、これもしかたなくアセンブラで記述しなくてはならないのだろうかと思われた。ところがどっこい、Cの柔軟性をあなどってはいけない。引数リストを2バイト単位にばらせばどんな引数でも受けられるというのが鍵である、10は2で割り切れるのだから。

具体的には、まずダミー引数を用意する。その名もずばり、DUMMY型（正体はただのintだが）。その引数dummyを指すポインタ&dummyを、2バイト整数の配列par[]へのポインタにキャストするのである。これでどんな引数が来ても大丈夫だ。引数のアクセスについてはマクロをしたまま使ったので、それほど関数本体では苦勞せずにすむだろう。しかし泥臭さには拭きたいものがある。

引数リストの構造などは説明すると長くなるし、マクロの使い方さえ理解すれば十分だと思うのでもうこれ以上は説明しないが、もっと詳しく知りたい方は、本文の最後に掲げた参考文献をご覧ください。親切かつエレガントな技法に出会えるであろう。

ると、途中でペイントが止まってしまうことがあるが、これはふつうのペイントでも状況は同じであろう。このペイントはかなりな好条件でないとい動いてくれない、わがままペイントルーチンなのであった、残念ながら。

タイルとトーン

スキャンコンバージョンaa_scanconv()とペイントaa_paint()では、タイルとトーンを使うことができる。使い方はZ'sSTAFFのタイル&トーンとほぼ同じで、描画の色にはカラーコード(単色)とタイルパターンのいずれかが選べ、トーンは使うか使わないかが選べる。

さらにスキャンコンバージョンでは、トーン指定の際に下地が透けて見えるか、それとも単に塗りつぶすのかを選ぶこともできる(ペイントは、そもそもアルゴリズム自体が下地の存在に大きく依存しているので、下地は透けて見えるのが当然なのである)。これにより、スクリーントーンを貼るのと同様の効果を狙っている。

ただし、下地が透けるモードでは、背景が黒いところにどんなトーンを張り付けてもなにも出ないので注意。スクリーント

ンもペイントと同様、白地に引いた黒い線で絵を描き、その上に貼るのが基本である。

タイルやトーンのパターンの登録の手順について。まず画面に基本パターンを描いておいて、それをtile_get()関数やtone_get()関数で取り込んで登録する。トーンは例によって、取り込んだパターンのうち、赤成分だけを見ている。まあモノトーンで描いておけば安心。また、通常のカラーコードでは明るい(白に近い)色のほうが値が大きいが、トーン登録に限っては、暗い(黒に近い)ほうがトーンの色が濃いとみなされる。これはZ'sSTAFFをまねたのだが、こちらのほうがわかりやすいようだ。

パターンを登録しておけば、ペイントでもスキャンコンバージョンでも、タイル番号やトーン番号を指定すれば呼び出すことができる。いろいろ指定して使い方を覚えていただきたい。

登録は本番の描画に先立ってやっておいたほうが、画面が乱れずにすむ。また一度登録しておけば、BASICを終わるまでパターンは消えないことになっている(BASICの変数とは別の場所に領域を確保している)。だから標準的なタイル/トーンパターンを定義するプログラムを描画プログラ

ムとは別に作り、BASIC起動時に一度だけrunしておけば、あとはそのパターンがずっと使える。

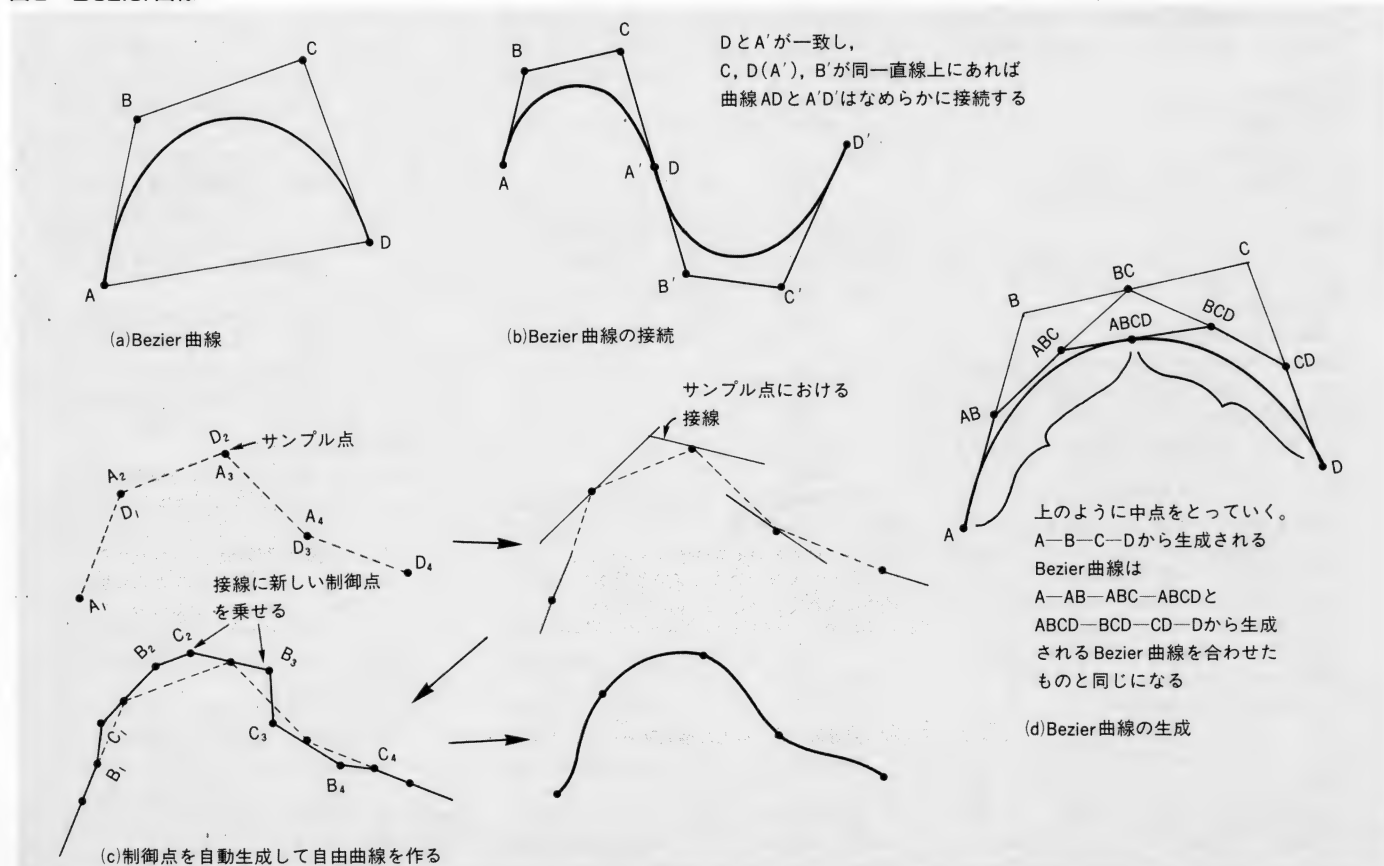
自由曲線

自由曲線にはBezier(ベジエ)曲線を使っている。Bezier曲線は4個ひと組の制御点を取る曲線である。その4つの制御点を順にA,B,C,Dとすると、Bezier曲線は次のような特徴を持つ。

- ・制御点Aから出発し、そこでは線分ABに接している。
- ・四角形ABCDの中に入る。
- ・制御点Dで終わり、そこでは線分CDに接している。

Bezier曲線は与えられた制御点すべてを通るわけではない(制御点BとCは通らない)が、これでは使いにくい。ふつうのユーザーなら、画面上にぽんぽんと点を置いて、その点を通る滑らかな曲線がほしい。かといっていちいち制御点BやCを手計算でつけ足していたのでは使いものにならない。下手な計算をすると、隣り合ったBezier曲線が滑らかにつながってくれないということになる。点列に記録されたすべての点を滑らかに通る曲線を生成するために

図6 Bezier曲線



は、制御点BやCをうまく自動的に計算して発生させる必要がある。

困っていたところで、以前に大学のコンピュータグラフィックの演習でうまくアルゴリズムを習ったことを思い出した。ここではそれを借用している。

その考え方を簡単に説明しておく。点列に記録されている点がサンプル点になる。まず各サンプル点上で、目的の曲線の接線を求める。次にその接線上に制御点BとCを乗せる。サンプル点は制御点AおよびDになる。こうしてできた制御点でBezier曲線を描かせると、曲線はサンプル点を通ってくれるし、しかもそのサンプル点上でなめらかにつながってくる。

最後に、制御点A～Dが与えられたときのBezier曲線の発生のしかたを説明しよう。制御点の midpoint どうしをつないでいって新しい制御点を発生する。新しい制御点は2組できるのだが、それら発生する2つのBezier曲線をつなぎ合わせると、求めるBezier曲線が得られるのである。2分割して統合するのだから、再帰が使える。再帰的に新しい制御点を発生し、十分制御点の間隔が短くなったところで再帰を打ち切る。そんな制御点なら、いきなり線分をつない

でしまっても、十分滑らかな曲線に見える。そのレベルまで再帰を続ければいいというわけ。

最後に

漫画家の道具はペンとインクと墨とスクリーンと、ほかになにがあるかはよく知らないが、その真似ごとを、ある程度のクオリティでできるようにはなったと思う。それでも動作テスト用のサンプルを作ろうとしてやっぱり嫌だなと思ったのは、どうしても数値を意識しておかないとなにも作れないので、いきおいつまらない図形で我慢してしまうところである。

このままでは人間デジタイザやドットナーの忍耐力が必要だ。マウスでばっばと描けるのが理想であろうが、それにはどうしてもマウスの動きに追従できるだけのレスポンスがいる。いっそアセンブラで全部書き下ろそうかと思ってしまうが、いまはコンパイラの手助けが頼りという状態だ。

ともあれ、アンチエイリアシングを手動ドット打ちなど使わずに実現できる可能性は示せたと思う。ジャギーフリーのグラフィックプリミティブを装備したペインティン

グツールというのはまだまだ先の話であろ
うが、その目標への第一歩としてこの外部関数をお使いいただければ幸いである。

参考文献

- (X-BASICの外部関数をCで書く方法について)
- ・C調言語講座PRO-68K第1回 まずはprintfより始めよ、祝 一平, Oh!X 1988年7月号, pp. 98-104
- ・C compiler PRO-68K プログラマーズマニュアル
- ・X 68000 BASIC入門 最終回 必殺サンプリング戦法, 中森 章, Oh!X 1988年7月号, pp. 129-136
- ・Oh!X質問箱, 村田 敏幸, Oh!X 1988年12月号, pp.129-167
- (幅のある線分について)
- ・アルゴリズムとプログラムによるコンピュータグラフィックス [I], S.Harrington著, 郡山彬訳, マグロウヒル, pp.32-33
- (ソーティングアルゴリズムについて)
- ・PascalとCプログラムによるアルゴリズムとデータ構造ハンドブック, G.H.Gonnet著, 玄光男・荒実・松本直文共訳, 啓学出版, pp.129-136
- (Bezier曲線の制御点を自動生成することについて)
- ・コンピュータ・グラフィックスの基礎, 鈴木賢次郎, 長島忍, 鈴木宏正, pp.A18-A20
- (Bezier曲線の再帰的分割による構成法について)
- ・アルゴリズムとプログラムによるコンピュータグラフィックス [II], S.Harrington著, 郡山彬訳, マグロウヒル, pp.539-543

表1 関数リファレンス

オーバーサンプリング倍率はソースリスト (anti.h) の OVERSAMPLE の値を書き換えることで変えることができるが、今回は8倍オーバーサンプリングとした。

点列のフォーマット

int pts (n, 2) で宣言する。nは格納できる点列の長さの最大値。

(ヘッダ情報)

pts (0, 0) ... 点列の長さ、頂点数 (≤n)

pts (0, 1) ... 点列のタイプ (0のとき片道通行, 1のとき循環する)

pts (0, 2) ... オーバーサンプリング倍数 (点列がオーバーサンプリング座標のときには8が入る)

(1 ≤ i ≤ pts (0, 0) なる頂点iの情報)

pts (i, 0) ... x座標

pts (i, 1) ... y座標

pts (i, 2) ... 幅 (オーバーサンプリング座標での値。これが8だと1ピクセル分の幅になる)

関数リファレンス

* どの関数にも、戻り値はない。

pts_oversample (pts)

(引数)

int pts (n, 2)

(機能)

通常のサンプリングレートで記述された点列 pts をオーバーサンプリング座標に変換する。

(注意)

点列がオーバーサンプリング座標かどうかは、pts (0, 2) の値で調べる。ここにオーバーサンプリング倍数 (8) が入っていれば、その点列はオーバーサンプリング座標である。

関数のうち、点列を引数にとるものは、オーバーサンプリング座標に変換しないと使えない。

pts_curve (pts1, w1, w2, pts2)

(引数)

int pts1 (n1, 2), w1, w2, pts2 (n2, 2)

(機能)

点列 pts1 の各頂点を通る自由曲線を生成し、点列 pts2 に格納する。その際、始点と終点での幅を w1, w2 とし、そのあいだの線の幅を線形補間する。

(注意)

曲線を微小線分で近似するので点列 pts2 は多少大きめに取る。

場合にもよるが、pts (1000, 2) 程度にしておけばよい。

配列の大きさが不足しているとエラーになる。

pts_append (pts1, pts2)

(引数)

int pts1 (n1, 2), pts2 (n2, 2)

(機能)

点列 pts1 と pts2 をつなげ、pts1 に格納する。

(注意)

pts1 の最後の点と pts2 の最初の点が一致するように pts2 を移動してからアペンドする。

pts1 の大きさ (n1の値) は、新しい点列の長さ、つまり (pts1 (0, 0) + pts2 (0, 0)) 以上用意しておくこと。

新しい点列のタイプ (片道通行か循環するか) は、もとの点列のうち pts1 のタイプにあわせる。

pts_move (pts1, x, y, pts2)

(引数)

int pts1 (n1, 2), x, y, pts2 (n2, 2)

(機能)

点列 pts1 をオフセット x, y で点列 pts2 に移動する。

(注意)

オフセットはオーバーサンプリング座標で指定すること (各座標を8倍する)。

オフセットを x, y 共に 0 とした場合は、点列コピーの役割も果たす。

aa_lines (pts, c)

(引数)

int pts(n, 2), c

(機能)

点列 pts に沿って、線の色 c で連続描画する。線の幅は各頂点に記録されている値を用いる。

(注意)

線の幅が太すぎると、表示が一部欠けることがある。

aa_scanconv(pts, cmode, c または n_tile, tmode, n_tone)

(引数)

int pts(n, 2), cmode, c, n_tile, tmode, n_tone

(機能)

点列 pts を輪郭とする多角形の内部を塗りつぶす。

cmode=0...c で指定される色 (単色) で塗る。

1...n_tile で指定されるタイルパターンで塗る。

tmode=0...トーンは用いない。下地の色と関係なく塗る。

1...n_tone で指定されるトーンを用いる。下地の色と関係なく塗る。

2...トーンは用いない。下地が透けて見える。

3...n_tone で指定されるトーンを用いる。下地が透けて見える。

(注意)

下地は、白地に黒い線を引いた場合を想定している。

下地のうち赤成分のみを取っている (青、緑成分は無視) ので、思いどおりの結果が出ないこともある。

tmode=2, 3 の場合は、黒い背景の場所に塗ってもなにも描画しない。

aa_paint(x, y, cmode, c または n_tile, tmode, n_tone)

(引数)

int x, y, cmode, c, n_tile, tmode, n_tone

(機能)

白地の中の点 (x, y) を出発点として、黒い線で囲まれた閉領域を塗りつぶす。

aa_lines () で描画されたような、境界のはっきりしない領域も塗る。色 (またはタイルパターン)、トーンは aa_scanconv () に準ずる。

(注意)

座標(x, y)には、オーバーサンプリングでないふつうの座標を指定すること。

境界判定アルゴリズムの関係上、あまり狭い部分を塗ることはできない。

白地から黒い境界に向かって塗るので、黒地から塗ることはできない。輝度ビットをペイントずみフラグとして用いている。

tile_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

タイル番号nのタイルパターンをグラフィック画面の(x1, y1)-(x2, y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には、オーバーサンプリングでないふつうの座標

を指定すること。

取り込めるパターンの大きさには制限がある (anti.h で定義されている T_SIZE の値)。

(x1, y1) が始点で (x2, y2) が終点。大小関係を変えれば、反転したパターンも取り込める。

tone_get(n, x1, y1, x2, y2)

(引数)

int n, x1, y1, x2, y2

(機能)

トーン番号nのトーンをグラフィック画面の(x1, y1)-(x2, y2)の領域から取り込む。

(注意)

座標(x1, y1), (x2, y2)には、オーバーサンプリングでないふつうの座標を指定すること。

取り込めるパターンの大きさには制限がある (anti.h で定義されている T_SIZE の値)。

(x1, y1) が始点で (x2, y2) が終点。大小関係を変えれば、反転したパターンも取り込める。

トーンの濃さは赤成分から取る。黒に近いほど (輝度が低いほど) 濃くなる。

whitepaper()

(引数)

なし

(機能)

白紙を作る

(注意)

fill (0, 0, 511, 511, 65534) と同じ。

輝度ビットが立っていないのに注意 (aa_paint () に支障をきたさないように)。

reverse()

(引数)

なし

(機能)

画面を反転させる

(注意)

どうしても黒地に aa_paint () で塗りつぶしたいときに使う。

いったん反転させて (白地になっている) 塗り、もう一度反転させて戻す。

反転に際しては輝度ビットをいじらない。

maskclear()

(引数)

なし

(機能)

輝度ビットをすべて 0 にする。

(注意)

aa_paint () は、輝度ビットの立っているところでは使えない。

先にこのマスクをクリアしておくための関数。

リスト3

===== aa_scanconv.c =====

```
1: /***** アンチエイリアシングつきソリッドスキャンコンバーション *****/
2:
3: #include <graph.h>
4: #include "anti.h"
5:
6: /* 多角形の辺 ... 線分の集合 */
7: typedef struct {
8:   int x, y, dx2, dy2, sx, e, ry;
9: } EDGE;
10:
11: /* リスト処理用の定数 */
12: #define FIRST 0 /* アクティブエッジリストの最初の要素 */
13: #define LAST 1 /* アクティブエッジリストの最後の要素 */
14: #define FORWARD 0 /* アクティブエッジリストの次要素 */
15: #define BACKWARD 1 /* アクティブエッジリストの前要素 */
16: #define NEXT 1 /* フリーリストの次要素 */
17: #define NIL -1 /* リストの終端を示す */
18:
19: #define MAXACTIVE 128 /* 1本のスキャンラインを切る辺の数 */
20: #define MAXEDGE 1024 /* 一度に処理できる辺の数 */
21:
22: EDGE edge[ MAXEDGE ];
23: EDGE *edgeptr[ MAXEDGE ], *activeedgeptr[ MAXACTIVE ];
24: int activelist[ MAXACTIVE ][2]; /* アクティブエッジリストは双方向リスト・フリーリストと共用 */
25: int scanlinebuffer[ MAXACTIVE ]; /* 輪郭とスキャンラインの交点の x 座標 */
26:
27: /* タイルおよびトーン (ペイントと共有) */
28: extern unsigned char Color[3]; /* R,G,B の 3色 */
29: extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ]; /* R,G,B の 3色 */
30: extern unsigned int Tile_x[ N_TILE ], Tile_y[ N_TILE ]; /* タイルパターンの大きさ */
31: extern unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ]; /* 単色 */
```

```
32: extern unsigned int Tone_x[ N_TILE ], Tone_y[ N_TILE ]; /* トーンの大きさ */
33:
34: extern unsigned int Alpha[ N_PIXEL ]; /* 1 スキャンラインふんの寄与率バフファ */
35: extern unsigned short Sbuf[ N_PIXEL ]; /* 1 スキャンラインふんのフレームバフファ */
36:
37: extern unsigned char OVERSAMPLE_NOTYET[];
38:
39: extern int tile_tone_check(); /* タイル・トーンの指定が正しいかどうか調べる */
40:
41: unsigned char TOO_COMPLEX[] = "入力した点列が複雑すぎます";
42:
43: FUNC aa_scanconv( dummy )
44: DUMMY dummy;
45: /* PTS pts, int cmode, c/n_tile, tmode, n_tone */
46: {
47:   PTS *pts;
48:   int cmode, c, n_tile, tmode, n_tone;
49:   int tile_x, tile_y, tone_x, tone_y;
50:
51:   int n_edge, y_min, y_max;
52:   int n, i, j, d, fw, bk, x, x1, x2, y, y1, y2, tmp;
53:   int active[2], inactive, free;
54:   EDGE *tmppt;
55:   unsigned int r, g, b, rl, gl, bl, v, vm, a, s;
56:
57:   ARGSET( dummy );
58:   ARYSZ(1);
59:   pts=PARTOP(1);
60:   cmode=IVALUE(2);
61:   c=n_tile=IVALUE(3);
62:   tmode=IVALUE(4);
63:   n_tone=IVALUE(5);
```



```

64:
65: /* 点列のタイプ、つまり pts[0][1] は無視される ( CYCLIC として扱われる ) */
66: npts[0][0];
67: if ( n>MAXEDGE ) {
68: #ifdef _GNUC
69: asm ( " lea.l _TOO_COMPLEX,a1" );
70: #else
71: #asm
72: lea.l _TOO_COMPLEX,a1
73: #endasm
74: #endif
75: return ( 1 );
76: }
77: if ( pts[0][2]!=OVERSAMPLE ) {
78: #ifdef _GNUC
79: asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
80: #else
81: #asm
82: lea.l _OVERSAMPLE_NOTYET,a1
83: #endasm
84: #endif
85: return ( 1 );
86: }
87:
88: if ( cmode==COLOR ) {
89: bl=BLUE( c );
90: rl=RED( c );
91: gl=GREEN( c );
92: }
93: itile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y );
94: if ( i==1 ) return ( 1 );
95:
96: y_min=N_PIXEL*OVERSAMPLE*2; y_max=-N_PIXEL*OVERSAMPLE*2;
97: for ( i=1, n_edges=0; i<n; i++ ) {
98: x1 = pts[i][0];
99: y1 = pts[i][1];
100: if ( i!=n ) { /* 通常の点 */
101: x2 = pts[i+1][0];
102: y2 = pts[i+1][1];
103: } else { /* 終点は始点とつなげる */
104: x2 = pts[1][0];
105: y2 = pts[1][1];
106: }
107: if ( y2==y1 ) continue; /* 水平なエッジは使わない */
108: if ( y2<y1 ) { /* スキャンコンバージョンは上から下へ処理する */
109: tmp=x1; x1=x2; x2=tmp;
110: tmp=y1; y1=y2; y2=tmp;
111: }
112: if ( y_min>y1 ) y_min=y1; /* 図形の存在範囲を求める */
113: if ( y_max<y2 ) y_max=y2;
114: edge[n_edge].x = x1; /* Bresenham アルゴリズムの初期値 */
115: edge[n_edge].y = y1;
116: edge[n_edge].dx2= 2*ABS( x2-x1 );
117: edge[n_edge].sx= SGN( x2-x1 );
118: edge[n_edge].dy2= 2*( y2-y1 );
119: edge[n_edge].ny= ( y2-y1 );
120: edge[n_edge].e = - ( y2-y1 );
121: edgeptr[n_edge]=(&edge[n_edge]);
122: n_edge++;
123: }
124: for ( d=n_edge; d>1; ) { /* エッジを始点が上にあるものから順にソートする */
125: if ( d<5 ) { /* Shell ソートを用いている */
126: d=1;
127: } else {
128: d=(5*d-1)/11;
129: }
130: for ( i=n_edge-1-d; i>=0; i-- ) {
131: tmp=edgeptr[i];
132: for ( j=i+d; j<=n_edge-1; j+=d ) { /* ポインタだけを並べ替えてスピードアップを図る */
133: if ( tmp->y<edgeptr[j]->y ) { /* アクティブでないエッジの先頭 */
134: edgeptr[j-d] = edgeptr[j];
135: edgeptr[j] = tmp;
136: }
137: }
138: }
139: free=0; /* フリーリストの先頭の要素 */
140: active[FIRST]=active[LAST]=NIL; /* アクティブエッジリストを空にする */
141: inactive=0; /* アクティブでないエッジの先頭 */
142: for ( i=0; i<MAXACTIVE-1; i++ ) { /* フリーリストを初期化する */
143: activelist[i][NEXT]=i+1;
144: }
145: activelist[MAXACTIVE-1][NEXT]=NIL; /* フリーリストの終端 */
146:
147: for ( y=y_min; y<=y_max; y++ ) {
148: /* 最初、またはスキャンラインの始めごとに、フレームバッファから取り込む */
149: if ( y==y_min || (y%OVERSAMPLE)==0 ) {
150: get( 0, PIX(y), N_PIXEL-1, PIX(y), Sbuf, N_PIXEL*sizeof(short) );
151: for ( x=0; x<N_PIXEL; x++ ) Alpha[x]=0;
152: }
153: /* 新しい active edge を探す (始点が現在のスキャンラインと重なるエッジ) */
154: while ( !inactive[n_edge] ) {
155: if ( edgeptr[inactive[n_edge]]->y != y ) break; /* ソート済みので打ち切り条件は */
156: if ( free==NIL ) { /* アクティブエッジリストの容量を超えた */
157: #ifdef _GNUC
158: asm ( " lea.l _TOO_COMPLEX,a1" );
159: #else
160: #asm
161: lea.l _TOO_COMPLEX,a1
162: #endasm
163: #endif
164: return ( 1 );
165: }
166: /* 新しいエッジをフリーリストから取ってくる */
167: if ( active[FIRST]==NIL ) { /* アクティブエッジリストが空の場合、新規に作る */
168: active[FIRST]=active[LAST]=free;
169: free=activelist[free][NEXT];
170: }
171: activelist[active[LAST]][FORWARD]=active[FIRST];
172: activelist[active[FIRST]][BACKWARD]=active[LAST];
173: }
174: else { /* アクティブエッジリストに新しいエッジを追加する */
175: fw=free;
176: free=activelist[free][NEXT];
177: activelist[active[LAST]][FORWARD]=fw;
178: activelist[fw][FORWARD]=active[LAST];
179: activelist[fw][BACKWARD]=active[FIRST];
180: active[LAST]=fw;
181: }
182: activeedgeptr[active[LAST]]=edgeptr[inactive];
183: inactive++;
184: }
185: /* アクティブエッジとスキャンラインとの交点を求める */
186: n=0;
187: for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
188: tmp=activeedgeptr[i];
189: if ( --(tmp->ry)<0 ) { /* 寿命の切れたエッジはリストから削除する */
190: fw=activelist[i][FORWARD]; /* 双方向リストは削除が楽 */
191: bk=activelist[i][BACKWARD];
192: activelist[i][NEXT]=free; /* 削除したエッジはフリーリストにつなぐ */
193: free=i;
194: if ( fw==NIL && bk==NIL ) {
195: active[FIRST]=active[LAST]=NIL;
196: continue;
197: }
198: if ( fw==NIL ) {

```



```

324:     y2 = PIX(pts[i+1][1]);
325:   } else {
326:     x2 = PIX(pts[i][0]);
327:     y2 = PIX(pts[i][1]);
328:   }
329:   if ( y2==y1 ) continue;
330:   if ( y2<y1 ) {
331:     tmpx=x1; x1=x2; x2=tmp;
332:     tmpy=y1; y1=y2; y2=tmp;
333:   }
334:   if ( y_min>y1 ) y_min=y1;
335:   if ( y_max<y2 ) y_max=y2;
336:   edge[n_edge].x = x1;
337:   edge[n_edge].y = y1;
338:   edge[n_edge].dx2 = 2*ABS( x2-x1 );
339:   edge[n_edge].sx = SGN( x2-x1 );
340:   edge[n_edge].dy2 = 2*( y2-y1 );
341:   edge[n_edge].ry = ( y2-y1 );
342:   edge[n_edge].e = -( y2-y1 );
343:   edgeptr[n_edge] = &edge[n_edge];
344:   n_edge++;
345: }
346: for ( d=n_edge; d<1; ) {
347:   if ( d<5 ) {
348:     d=1;
349:   } else {
350:     d=(5*d-1)/11;
351:   }
352:   for ( i=n_edge-1-d; i>=0; i-- ) {
353:     tmp=edgeptr[i];
354:     for ( j=i+d; j<(n_edge-1) && ((tmp->y)>edgeptr[j]->y); j+=d ) {
355:       edgeptr[j-d] = edgeptr[j];
356:     }
357:     edgeptr[j-d] = tmp;
358:   }
359: }
360: free=0;
361: active[FIRST]=active[LAST]=NIL;
362: inactive=0;
363: for ( i=0; i<MAXACTIVE-1; i++ ) {
364:   activelist[i][NEXT]=i+1;
365: }
366: activelist[MAXACTIVE-1][NEXT]=NIL;
367:
368: for ( y=y_min; y<cy_max; y++ ) {
369:   while ( !inactive[n_edge] ) {
370:     if ( edgeptr[inactive]->y != y ) break;
371:     if ( free==NIL ) {
372: #ifdef _GNUC
373:       asm ( " lea.l _TOO_COMPLEX,a1" );
374: #else
375:       #asm
376:       lea.l _TOO_COMPLEX,a1
377: #endif
378:       return ( 1 );
379:     }
380:   }
381:   if ( active[FIRST]==NIL ) {
382:     active[FIRST]=active[LAST]=free;
383:     free=activelist[free][NEXT];
384:     activelist[active[LAST]][FORWARD]=activelist[active[FIRST]][BACKWARD]=NIL;
385:   } else {
386:     free=free;
387:     free=activelist[free][NEXT];
388:     activelist[active[LAST]][FORWARD]=f;

```

```

369:     activelist[f][FORWARD]=NIL;
390:     activelist[f][BACKWARD]=active[LAST];
391:     active[LAST]=f;
392:   }
393:   activeedgeptr[active[LAST]]=edgeptr[inactive];
394:   inactive++;
395: }
396: n=0;
397: for ( i=active[FIRST]; i!=NIL; i=activelist[i][FORWARD] ) {
398:   tmp=activeedgeptr[i];
399:   if ( (tmp->ry)<0 ) {
400:     f=activelist[i][FORWARD];
401:     b=activelist[i][BACKWARD];
402:     activelist[i][NEXT]=free;
403:     free=f;
404:     if ( f==NIL && b==NIL ) {
405:       active[FIRST]=active[LAST]=NIL;
406:       continue;
407:     }
408:     if ( f==NIL ) {
409:       active[LAST]=b;
410:       activelist[b][FORWARD]=NIL;
411:       continue;
412:     }
413:     if ( b==NIL ) {
414:       active[FIRST]=f;
415:       activelist[f][BACKWARD]=NIL;
416:       continue;
417:     }
418:     activelist[b][FORWARD]=f;
419:     activelist[f][BACKWARD]=b;
420:     continue;
421:   }
422:   scanlinebuffer[n++] = tmp->x;
423:   (tmp->e) += (tmp->dx2);
424:   while ( (tmp->e) >= 0 ) {
425:     (tmp->x) += (tmp->sx);
426:     (tmp->e) -= (tmp->dy2);
427:   }
428: }
429: if ( n==0 ) continue;
430: scanlinebuffer[n]=N_PIXEL*2;
431: for ( i=n-2; i>=0; i-- ) {
432:   tmp=scanlinebuffer[i];
433:   for ( j=i+1; tmp<scanlinebuffer[j]; j++ ) {
434:     scanlinebuffer[j-1]=scanlinebuffer[j];
435:   }
436:   scanlinebuffer[j-1]=tmp;
437: }
438: for ( i=0; i<n-1; i+=2 ) {
439:   line[ scanlinebuffer[i], y, scanlinebuffer[i+1], y, c, 0xFFFF ];
440: }
441: }
442: return ( 0 );
443: }

```

リスト4

```

----- aa_lines.c -----
1: /***** アンチエイシング付き点列描画 *****/
2: #include <graph.h>
3: #include <math.h>
4: #include "anti.h"
5: extern unsigned char Color[3]; /* 描画色 */
6: extern unsigned int Alpha[ N_PIXEL ]; /* 1ラインぶんの寄与率バツファ */
7: extern unsigned short Sibuf[ N_PIXEL ]; /* 1ラインぶんのフレームバツファ */
8: extern unsigned char OVERSAMPLE_NOTYET[];
9: int Direction; /* 描画中の点列の傾きが大きい小さいか */
10: void aa_line( x1, y1, x2, y2, w ) /* 1区間だけ (横分1本) 描く */
11: int x1, y1, x2, y2, w;
12: {
13:   double h0;
14:   int newdirection, i, j, x, y, sx, dx, dx2, sy, dy, dy2, e, h;
15:   int xa, xb, ya, yb;
16:   unsigned int r, g, b;
17:   unsigned int a;
18:   unsigned short c, s;
19:   x=x1; /* Bresenhamアルゴリズムの初期設定 */
20:   y=y1;
21:   dx=ABS( x2-x1 );
22:   sx=SGN( x2-x1 );
23:   dx2=dx*2;
24:   dy=ABS( y2-y1 );
25:   sy=SGN( y2-y1 );
26:   dy2=dy*2;
27:   if ( dx==0 && dy==0 ) return;
28:   e=0;
29:   c=RGB( Color[1], Color[2], Color[0] );
30:   if ( dx>dy ) { /* 傾きが小さい場合は x でループ */
31:     if ( sx>0 ) {
32:       newdirection=1;
33:     } else {
34:       newdirection=3;
35:     }
36:   } else { /* 傾きが大きい場合は y でループ */
37:     if ( sy>0 ) {
38:       newdirection=2;
39:     } else {
40:       newdirection=4;
41:     }
42:   }
43:   if ( newdirection==1 || newdirection==3 ) { /* 傾きが小さい場合は x でループ */
44:     /* Direction が切り替わるとき、スクリーンから1ライン取り込む */
45:     if ( Direction != newdirection ) { /* Direction=0 (最初) の場合も含む */
46:       get( PIX(x), 0, PIX(x), N_PIXEL-1, Sibuf, N_PIXEL*sizeof(short) );
47:       for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0; /* 寄与率バツファをクリアする */
48:       Direction=newdirection;
49:     }
50:     /* 幅から高さを計算する */
51:     h0 = ( (double)w * sqrt((double)dx*(double)dx+(double)dy*(double)dy) ) / dx;
52:     h = (int)h0;
53:     y = (int)(h0/2.0); /* 下端線の始点 */
54:     for ( i=0; i<dx; i++, x+=sx ) { /* メインループ */
55:       /* 1ライン処理することによって新しいラインを取り込む */
56:       /* 始点ではラインを取り込むとは限らない (前の線分の情報を残す) */
57:       if ( (sx==1 && SUBPIX(x)==0) || (sx==-1 && SUBPIX(x)==(OVERSAMPLE-1)) ) {
58:         get( PIX(x), 0, PIX(x), N_PIXEL-1, Sibuf, N_PIXEL*sizeof(short) );

```

```

59:         for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0; /* 寄与率バツファをクリアする */
60:       }
61:       ya=y; yb=y+h-1; /* 寄与率を更新する */
62:       if ( yb==0 && ya<(N_PIXEL*OVERSAMPLE) ) {
63:         if ( ya<0 ) ya=0;
64:         if ( yb>(N_PIXEL*OVERSAMPLE) ) yb=(N_PIXEL*OVERSAMPLE-1);
65:         if ( PIX(ya) == PIX(yb) ) {
66:           Alpha[PIX(ya)] += (yb-ya);
67:         } else {
68:           Alpha[PIX(ya)] += (OVERSAMPLE-SUBPIX(ya));
69:           for ( j=(PIX(ya)+1); j<PIX(yb); j++ ) {
70:             Alpha[j] += OVERSAMPLE;
71:           }
72:           Alpha[PIX(yb)] += (SUBPIX(yb)+1);
73:         }
74:       }
75:       e+=dy2; /* Bresenham アルゴリズムで下端線分を発生する */
76:       if ( e>dx ) {
77:         y+=sy;
78:         e-=dx2;
79:       }
80:       /* 1ラインごと (または終点) でa合成と出力を行う */
81:       if ( i==dx-1 || (sx==-1 && SUBPIX(x)==0) || (sx==1 && SUBPIX(x)==(OVERSAMPLE-1)) ) {
82:         for ( j=0; j<N_PIXEL; j++ ) {
83:           if ( (a=Alpha[j])>0 ) continue;
84:           if ( a>OVER2 ) { /* 寄与率1ならベータ書き */
85:             Sibuf[j] = c;
86:           } else { /* そうでないならa合成 */
87:             s=Sibuf[j];
88:             b=( (OVER2-a)*BLUE(s) + a*Color[0] ) / OVER2;
89:             r=( (OVER2-a)*RED(s) + a*Color[1] ) / OVER2;
90:             g=( (OVER2-a)*GREEN(s) + a*Color[2] ) / OVER2;
91:             Sibuf[j] = RGB( r, g, b );
92:           }
93:         }
94:         put( PIX(x), 0, PIX(x), N_PIXEL-1, Sibuf, N_PIXEL*sizeof(short) );
95:       }
96:     }
97:   } else { /* 傾きが大きい場合は y でループ、以下同様 */
98:     if ( Direction != newdirection ) {
99:       get( 0, PIX(y), N_PIXEL-1, PIX(y), Sibuf, N_PIXEL*sizeof(short) );
100:       for ( i=0; i<N_PIXEL; i++ ) Alpha[i]=0;
101:       Direction=newdirection;
102:     }
103:     h0 = ( (double)w * sqrt((double)dx*(double)dx+(double)dy*(double)dy) ) / dy;
104:     h = (int)h0;
105:     x = (int)(h0/2.0);
106:     for ( i=0; i<dy; i++, y+=sy ) {
107:       if ( (sy==1 && SUBPIX(y)==0) || (sy==-1 && SUBPIX(y)==(OVERSAMPLE-1)) ) {
108:         get( 0, PIX(y), N_PIXEL-1, PIX(y), Sibuf, N_PIXEL*sizeof(short) );
109:         for ( j=0; j<N_PIXEL; j++ ) Alpha[j]=0;
110:       }
111:       xa=x; xb=x+h-1;
112:       if ( xb>0 && xa<(N_PIXEL*OVERSAMPLE) ) {
113:         if ( xa<0 ) xa=0;
114:         if ( xb>(N_PIXEL*OVERSAMPLE) ) xb=(N_PIXEL*OVERSAMPLE-1);
115:         if ( PIX(xa) == PIX(xb) ) {
116:           Alpha[PIX(xa)] += (xb-xa);
117:         } else {

```



```

118: Alpha[PIX(xa)] += (OVERSAMPLE-SUBPIX(xa));
119: for ( j=(PIX(xa)+1); j<PIX(xb); j++) {
120:   Alpha[j] += OVERSAMPLE;
121: }
122: Alpha[PIX(xb)] += (SUBPIX(xb)+1);
123: }
124: }
125: e+=dx2;
126: if ( e>dy ) {
127:   x+=sx;
128:   e-=dy2;
129: }
130: if ( i==dy-1 || (say==1 && SUBPIX(y)==0) || (say==1 && SUBPIX(y)==(OVERSAMPLE-1))) {
131:   for ( j=0; j<N_PIXEL; j++) {
132:     if ( (a=Alpha[j])>0) continue;
133:     if ( a>OVER2 ) {
134:       S1buf[j]=0;
135:     } else {
136:       a=S1buf[j];
137:       b=( OVER2-a)*BLUE(s) + a*Color[0] /OVER2;
138:       r=( OVER2-a)*RED(s) + a*Color[1] /OVER2;
139:       g=( OVER2-a)*GREEN(s)+ a*Color[2] /OVER2;
140:       S1buf[j] = RGB( r, g, b );
141:     }
142:   }
143:   put( 0, PIX(y), N_PIXEL-1, PIX(y), S1buf, N_PIXEL*sizeof(short) );
144: }
145: }
146: }
147: return;
148: }
149: FUNC aa_lines( dummy ) /* 関数本体、全点列を描画する */
150: DUMMY dummy;
151: /* PTS *pts, int c */
152: {
153:   PTS *pts;
154:   int c;
155:   int i, n;
156:   ARGSET( dummy );
157:   ARYSET(1);
158:   pts=PARTOP(1);
159:   c=IVALUE(2);
160:   Color[0]=BLUE(c);
161:   Color[1]=RED(c);
162:   Color[2]=GREEN(c);
163:   npts[0][0];
164:   if ( pts[0][2]!=OVERSAMPLE ) {
165: #ifdef _GNU_

```

```

166:   asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
167: } else
168: #asm
169:   lea.l _OVERSAMPLE_NOTYET,a1
170: #endasm
171: #endif
172:   return ( 1 );
173: }
174: Directions=0; /* 始めは傾きの方向不明 */
175: for ( i=1; i<n; i++) {
176:   aa_line( pts[i][0], pts[i][1], pts[i+1][0], pts[i+1][1], pts[i][2] );
177: }
178: if ( pts[0][1]==CYCLIC ) { /* 点列が循環している場合、終点と始点をつなぐ */
179:   aa_line( pts[n][0], pts[n][1], pts[1][0], pts[1][1], pts[n][2] );
180: }
181:   return ( 0 );
182: }
183: FUNC lines( dummy ) /* アンチエイリアシングなしのバージョン */
184: DUMMY dummy;
185: /* PTS *pts, int c */
186: {
187:   PTS *pts;
188:   int c;
189:   int i, n;
190:   ARGSET( dummy );
191:   ARYSET(1);
192:   pts=PARTOP(1);
193:   c=IVALUE(2);
194:   npts[0][0];
195:   if ( pts[0][2]!=OVERSAMPLE ) {
196: #ifdef _GNU_
197:   asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
198: } else
199: #asm
200:   lea.l _OVERSAMPLE_NOTYET,a1
201: #endasm
202: #endif
203:   return ( 1 );
204: }
205: for ( i=1; i<n; i++) {
206:   line( PIX(pts[i][0]), PIX(pts[i][1]), PIX(pts[i+1][0]), PIX(pts[i+1][1]), c, 0xFFFF );
207: }
208: if ( pts[0][1]==CYCLIC ) {
209:   line( PIX(pts[n][0]), PIX(pts[n][1]), PIX(pts[1][0]), PIX(pts[1][1]), c, 0xFFFF );
210: }
211:   return ( 0 );
212: }

```

リスト5

```

===== aa_paint.c =====
1: /***** アンチエイリアシング対応イベントルーチン *****/
2: #include <graph.h>
3: #include "anti.h"
4: /* キュー（待ち行列、または FIFO バッファ） */
5: #define MAXQUEUE 1024
6: short Qx[ MAXQUEUE ], Qy[ MAXQUEUE ];
7: int QPi, QPo;
8: #define INIT_QUEUE ( QPi=QPo=0; )
9: #define ENQUEUE( X, Y ) { Qx[ QPi ]=(X); Qy[ QPi ]=(Y); QPi=(++QPi)%MAXQUEUE; }
10: #define DEQUEUE( X, Y ) { (X)=Qx[ QPo ]; (Y)=Qy[ QPo ]; QPo=(++QPo)%MAXQUEUE; }
11: #define EMPTY_QUEUE ( QPi=QPo=0; )
12: /* タイルおよびトーン（ソリッドスキャンコンバージョンと共有） */
13: extern unsigned char Color[3]; /* R,G,B の 3色 */
14: extern unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ]; /* R,G,B の 3色 */
15: extern unsigned int Tile_x[ N_TILE ], Tile_y[ N_TILE ]; /* タイルパターンの大きさ */
16: extern unsigned char Tone[ N_TONE ][ T_SIZE ]; /* 単色 */
17: extern unsigned int Tone_x[ N_TILE ], Tone_y[ N_TILE ]; /* トーンの大きさ */
18: extern int tile_tone_check(); /* タイル・トーンの指定が正しいかどうか調べる */
19: extern unsigned short S1buf[ N_PIXEL ]; /* 1 スキャンライン分のフレームバッファ */
20: unsigned short slbuf[512], slmbuf[512], sldbuf[512]; /* イベント用スキャンラインバッファ */
21: unsigned char _OUTOF_SCREEN[] = "指定した座標がスクリーンの範囲外です";
22:
23: FUNC aa_paint( dummy ) /* 関数本体 */
24: DUMMY dummy;
25: /* int x0, y0, c */
26: {
27:   int x0, y0, cmode, c, n_tile, tmode, n_tone;
28:   int tile_x, tile_y, tone_x, tone_y;
29:
30:   int i, x, y, x1, x2;
31:   int sign, sign1;
32:   unsigned int r, g, b, rl, gl, bl, v, vm, s;
33:
34:   ARGSET( dummy );
35:   ARYSET(1);
36:   x0=IVALUE(1);
37:   y0=IVALUE(2);
38:   cmode=IVALUE(3);
39:   c=n_tile=IVALUE(4);
40:   tmode=IVALUE(5);
41:   n_tone=IVALUE(6);
42:   if ( x0<0 || x0>N_PIXEL || y0<0 || y0>N_PIXEL ) { /* 画面外は塗れない */
43: #ifdef _GNU_
44:   asm ( " lea.l _OUTOF_SCREEN,a1" );
45: } else
46: #asm
47:   lea.l _OUTOF_SCREEN,a1
48: #endasm
49: #endif
50:   return ( 1 );
51: }
52:
53: if ( cmode==COLOR ) { /* タイルパターンを使わないなら描画色は一定 */
54:   bl=BLUE( c );
55:   rl=RED( c );
56:   gl=GREEN( c );
57: }
58:
59: itile_tone_check( cmode, n_tile, &tile_x, &tile_y, tmode, n_tone, &tone_x, &tone_y );
60: if ( i==1 ) return ( 1 );
61:
62: INIT_QUEUE; /* FIFO バッファを空にする */
63: ENQUEUE( x0, y0 ); /* シードを FIFO バッファに入れる */
64: vm=VALUE( point( x0, y0 ) ); /* 素点の最大値は出発点を100%とする */
65: if ( tmode&ON ) { /* トーンを考慮した素点の最大値 */
66:   vm=IMAX;
67: }
68:
69: while ( !EMPTY_QUEUE ) { /* イベント終了条件 */
70:   DEQUEUE( x0, y0 ); /* シードを FIFO バッファから取り出す */
71:   if ( point( x0, y0 )&FMASK ) continue; /* イベント済み */
72:   if ( point( x0, y0 )&MASK == 0 ) continue; /* 黒いところには塗っても仕方がない */
73:   if ( y0>0 ) /* 上への到達可能性を調べるためのバッファ */
74:     get( 0, y0-1, N_PIXEL-1, y0-1, slbuf, N_PIXEL*sizeof(short) );

```

```

75:   get( 0, y0, N_PIXEL-1, y0, slbuf, N_PIXEL*sizeof(short) );
76:   if ( y0<N_PIXEL-1 ) /* 下への到達可能性を調べるためのバッファ */
77:     get( 0, y0+1, N_PIXEL-1, y0+1, sldbuf, N_PIXEL*sizeof(short) );
78:
79:   for ( x=0; x<N_PIXEL; x++) {
80:     slbuf[x] = SLMASK;
81:     slmbuf[x] = S1buf[x]&SLMASK; /* 左右への到達可能性を調べるバッファ */
82:     sldbuf[x] = SLMASK;
83:     if ( slbuf[x]==0 ) slbuf[x]=FMASK; /* 黒いところには塗っても仕方がないので */
84:     if ( slmbuf[x]==0 ) slmbuf[x]=FMASK; /* 始めからイベント済みしておく */
85:     if ( sldbuf[x]==0 ) sldbuf[x]=FMASK;
86:   }
87:   for ( x1=x0; x1>=0; x1-- ) { /* 左へ塗り進める */
88:     if ( cmode==TILE ) { /* タイルパターン */
89:       bl=Tile[n_tile][y0%tile_y][x1%tile_x][0];
90:       rl=Tile[n_tile][y0%tile_y][x1%tile_x][1];
91:       gl=Tile[n_tile][y0%tile_y][x1%tile_x][2];
92:     }
93:     if ( tmode&ON ) { /* トーンあり */
94:       vm=(slmbuf[x1]>>VSHIFT)*Tone[n_tone][y0%tone_y][x1%tone_x];
95:     } else { /* トーンなし */
96:       vm=slmbuf[x1]>>VSHIFT;
97:     }
98:     if ( tmode&TP ) { /* 下地が透けて見える */
99:       s=S1buf[x1];
100:       b=( BLUE(s)*(vm-v) + bl*v )/vm;
101:       r=( RED(s)*(vm-v) + rl*v )/vm;
102:       g=( GREEN(s)*(vm-v) + gl*v )/vm;
103:     } else { /* 下地は無視 */
104:       b=bl*v/vm;
105:       r=rl*v/vm;
106:       g=gl*v/vm;
107:     }
108:     S1buf[x1]=RGBI( r, g, b, FMASK ); /* イベント済みフラグを立てる */
109:     if ( x1==0 ) break; /* さらに左に進めるか調べる */
110:     if ( slmbuf[x1-1]&FMASK ) break; /* イベント済みのところで止める */
111:     if ( slmbuf[x1]<slmbuf[x1-1] ) break; /* 次が明るくなりそうなら止める */
112:   }
113:   for ( x2=x0; x2<N_PIXEL; x2++) { /* 右へ塗り進める、以下同様 */
114:     if ( cmode==TILE ) {
115:       bl=Tile[n_tile][y0%tile_y][x2%tile_x][0];
116:       rl=Tile[n_tile][y0%tile_y][x2%tile_x][1];
117:       gl=Tile[n_tile][y0%tile_y][x2%tile_x][2];
118:     }
119:     if ( tmode&ON ) {
120:       vm=(slmbuf[x2]>>VSHIFT)*Tone[n_tone][y0%tone_y][x2%tone_x];
121:     } else {
122:       vm=slmbuf[x2]>>VSHIFT;
123:     }
124:     if ( tmode&TP ) {
125:       s=S1buf[x2];
126:       b=( BLUE(s)*(vm-v) + bl*v )/vm;
127:       r=( RED(s)*(vm-v) + rl*v )/vm;
128:       g=( GREEN(s)*(vm-v) + gl*v )/vm;
129:     } else {
130:       b=bl*v/vm;
131:       r=rl*v/vm;
132:       g=gl*v/vm;
133:     }
134:     S1buf[x2]=RGBI( r, g, b, FMASK );
135:     if ( x2==N_PIXEL-1 ) break;
136:     if ( slmbuf[x2+1]&FMASK ) break;
137:     if ( slmbuf[x2]<slmbuf[x2+1] ) break;
138:   }
139:   if ( y0>0 ) { /* 上へ塗り進める可能性を調べる */
140:     /* sign は傾きの方向 */
141:     /* 0 と 1 のスキャンラインの傾度が最大になる (sign が + か - に転じる) ところで塗り進める */
142:     /* 塗り進めない傾度が途中から出た場合には、その直前で塗り進める */
143:     sign=1;
144:     for ( x=x1; x<=x2; x++) {
145:       if ( slbuf[x]&FMASK || slmbuf[x]<slmbuf[x+1] ) {
146:         sign=-1;
147:         continue;
148:       }
149:       if ( x==x2 || slbuf[x+1]&FMASK || slmbuf[x+1]<slmbuf[x+1] ) {

```



```

150:         if ( sign>0 ) ENQUEUE( x, y0-1 );
151:     continue;
152: }
153: sign1=sbuf[x+1]-sbuf[x];
154: if ( sign>0 && sign1<0 ) ENQUEUE( x, y0-1 );
155: if ( sign=sign1<0 ) sign=sign1;
156: }
157: if ( y0<N_PIXEL-1 ) ( /* 下へ塗り進める可能性を調べる、以下同様 */
158:     sign1=
159:     for ( x=x1; x<=x2; x++ ) {
160:         if ( sdbuf[x]&PMASK || slmbuf[x]<sdbuf[x] ) {
161:             sign1=
162:             continue;
163:         }

```

```

164:     }
165:     if ( x==x2 || sdbuf[x+1]&PMASK || slmbuf[x+1]<sdbuf[x+1] ) {
166:         if ( sign>0 ) ENQUEUE( x, y0+1 );
167:         continue;
168:     }
169:     sign1=sdbuf[x+1]-sdbuf[x];
170:     if ( sign>0 && sign1<0 ) ENQUEUE( x, y0+1 );
171:     if ( sign=sign1<0 ) sign=sign1;
172: }
173: }
174: put( 0, y0, N_PIXEL-1, y0, Sbuf, N_PIXEL*sizeof(short) );
175: }
176: return ( 0 );
177: }

```

リスト6

```

===== aa_proc.c =====
1: /****** アンチエイリアシング関係の処理 (タイル・トーンなど) *****/
2: #include "anti.h"
3: /* 色 (カラーモードまたはタイルパターン) */
4: unsigned char Color[3]; /*R,G,Bの3色*/
5: unsigned char Tile[ N_TILE ][ T_SIZE ][ T_SIZE ]; /*R,G,Bの3色*/
6: unsigned int Tile_x[ N_TILE ], Tile_y[ N_TILE ]; /*タイルパターンの大きさ*/
7:
8: /* トーン */
9: unsigned char Tone[ N_TONE ][ T_SIZE ][ T_SIZE ]; /*単色*/
10: unsigned int Tone_x[ N_TONE ], Tone_y[ N_TONE ]; /*トーンの大きさ*/
11: unsigned int Alpha[ N_PIXEL ]; /*1スキャンラインおんの書き率ハフファ*/
12: unsigned short Sbuf[ N_PIXEL ]; /*1スキャンラインおんのフレームハフファ*/
13: unsigned short Temp[ T_SIZE*T_SIZE ]; /*tile.get(), tone.get() 用のテンポラリ配列*/
14: unsigned char TOOMANY_PATTERN[]="パターンの番号が大きすぎます";
15: unsigned char OVERSIZE[]="パターンのサイズが大きすぎます";
16:
17: FUNC tile_get( dummy )
18: DUMMY dummy;
19: /* int n, int x1, int y1, int x2, int y2 */
20: {
21:     int n, x1, y1, x2, y2;
22:     int i, j, dx, dy, sx, sy, x, y;
23:     unsigned short c;
24:     ARGSET( dummy );
25:     n=IVALUE(1);
26:     x1=IVALUE(2);
27:     y1=IVALUE(3);
28:     x2=IVALUE(4);
29:     y2=IVALUE(5);
30:     if ( n>N_TILE ) {
31:         #ifdef _GNUC_
32:             asm( " lea.l _TOOMANY_PATTERN,a1" );
33:         #else
34:             #asm
35:             lea.l _TOOMANY_PATTERN,a1
36:             #endasm
37:         #endif
38:         return ( 1 );
39:     }
40:     dx = ABS( x2-x1 )+1;
41:     dy = ABS( y2-y1 )+1;
42:     sx = SGN( x2-x1 );
43:     sy = SGN( y2-y1 );
44:     if ( dx>T_SIZE || dy>T_SIZE ) {
45:         #ifdef _GNUC_
46:             asm( " lea.l _OVERSIZE,a1" );
47:         #else
48:             #asm
49:             lea.l _OVERSIZE,a1
50:             #endasm
51:         #endif
52:         return ( 1 );
53:     }
54:     Tile_x[n] = dx;
55:     Tile_y[n] = dy;
56:     get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Temp, dx*dy*sizeof(short) );
57:     for ( i=0; i<=(sx>0)?0:(dx-1); i++) {
58:         for ( j=0; j<=(sy>0)?0:(dy-1); j++) {
59:             c=Temp[ydx*xj];
60:             Tile[n][i][j][0]=BLUE(c);
61:             Tile[n][i][j][1]=RED(c);
62:             Tile[n][i][j][2]=GREEN(c);
63:         }
64:     }
65:     return ( 0 );
66: }
67:
68: FUNC tone_get( dummy )
69: DUMMY dummy;
70: /* int n, int x1, int y1, int x2, int y2 */
71: {
72:     int n, x1, y1, x2, y2;
73:     int i, j, dx, dy, sx, sy, x, y;
74:     unsigned short c;
75:
76:     ARGSET( dummy );
77:     n=IVALUE(1);
78:     x1=IVALUE(2);
79:     y1=IVALUE(3);
80:     x2=IVALUE(4);
81:     y2=IVALUE(5);
82:     if ( n>N_TONE ) {
83:         #ifdef _GNUC_
84:             asm( " lea.l _TOOMANY_PATTERN,a1" );
85:         #else
86:             #asm
87:             lea.l _TOOMANY_PATTERN,a1
88:             #endasm
89:         #endif
90:         return ( 1 );
91:     }
92: }
93:
94: dx = ABS( x2-x1 )+1;
95: dy = ABS( y2-y1 )+1;
96: sx = SGN( x2-x1 );
97: sy = SGN( y2-y1 );
98:
99: if ( dx>T_SIZE || dy>T_SIZE ) {
100:     #ifdef _GNUC_
101:         asm( " lea.l _OVERSIZE,a1" );
102:     #else
103:         #asm
104:         lea.l _OVERSIZE,a1
105:         #endasm
106:     #endif
107:     return ( 1 );
108: }

```

```

109: Tone_x[n] = dx;
110: Tone_y[n] = dy;
111: get( MIN(x1,x2), MIN(y1,y2), MAX(x1,x2), MAX(y1,y2), Temp, dx*dy*sizeof(short) );
112: for ( i=0; i<=(sx>0)?0:(dx-1); i++) {
113:     for ( j=0; j<=(sy>0)?0:(dy-1); j++) {
114:         c=Temp[ydx*xj];
115:         Tone[n][i][j]=(IMAX-RED(c)); /* 黒い部分ほど濃くする */
116:     }
117: }
118: return ( 0 );
119: }
120: unsigned char ILLEGAL_NTILE[]="タイルパターン番号が不正です";
121: unsigned char TILE_TOO_LARGE[]="タイルパターンのサイズが大きすぎます";
122: unsigned char ILLEGAL_CMODE[]="色/タイルのモードを正しく指定して下さい";
123: unsigned char ILLEGAL_NTONE[]="トーン番号が不正です";
124: unsigned char TONE_TOO_LARGE[]="トーンのサイズが大きすぎます";
125: unsigned char ILLEGAL_TMODE[]="トーンのモードを正しく指定して下さい";
126: int tile_tone_check( cmode, n_tile, tile_x, tile_y, tmode, n_tone, tone_x, tone_y );
127: int cmode, n_tile, *tile_x, *tile_y, tmode, n_tone, *tone_x, *tone_y;
128: {
129:     switch ( cmode ) {
130:         case COLOR:
131:             break;
132:         case TILE:
133:             if ( n_tile >= N_TILE ) {
134:                 #ifdef _GNUC_
135:                     asm( " lea.l _ILLEGAL_NTILE,a1" );
136:                 #else
137:                     #asm
138:                     lea.l _ILLEGAL_NTILE,a1
139:                     #endasm
140:                 #endif
141:                 return ( 1 );
142:             }
143:             *tile_x=Tile_x[n_tile];
144:             *tile_y=Tile_y[n_tile];
145:             if ( *tile_x>T_SIZE || *tile_y>T_SIZE ) {
146:                 #ifdef _GNUC_
147:                     asm( " lea.l _TILE_TOO_LARGE,a1" );
148:                 #else
149:                     #asm
150:                     lea.l _TILE_TOO_LARGE,a1
151:                     #endasm
152:                 #endif
153:                 return ( 1 );
154:             }
155:             break;
156:         default:
157:             #ifdef _GNUC_
158:                 asm( " lea.l _ILLEGAL_CMODE,a1" );
159:             #else
160:                 #asm
161:                 lea.l _ILLEGAL_CMODE,a1
162:                 #endasm
163:             #endif
164:             return ( 1 );
165:             break;
166:     }
167:     switch ( tmode ) {
168:         case ON_TP:
169:             case ON_NTP:
170:                 if ( n_tone >= N_TONE ) {
171:                     #ifdef _GNUC_
172:                         asm( " lea.l _ILLEGAL_NTONE,a1" );
173:                     #else
174:                         #asm
175:                         lea.l _ILLEGAL_NTONE,a1
176:                         #endasm
177:                     #endif
178:                     return ( 1 );
179:                 }
180:                 *tone_x=Tone_x[n_tone];
181:                 *tone_y=Tone_y[n_tone];
182:                 if ( *tone_x>T_SIZE || *tone_y>T_SIZE ) {
183:                     #ifdef _GNUC_
184:                         asm( " lea.l _TONE_TOO_LARGE,a1" );
185:                     #else
186:                         #asm
187:                         lea.l _TONE_TOO_LARGE,a1
188:                         #endasm
189:                     #endif
190:                     return ( 1 );
191:                 }
192:                 break;
193:             case OFF_TP:
194:             case OFF_NTP:
195:                 break;
196:             default:
197:                 #ifdef _GNUC_
198:                     asm( " lea.l _ILLEGAL_TMODE,a1" );
199:                 #else
200:                     #asm
201:                     lea.l _ILLEGAL_TMODE,a1
202:                     #endasm
203:                 #endif
204:                 return ( 1 );
205:                 break;
206:     }
207:     return ( 0 );
208: }
209:
210: FUNC whitepaper()
211: {
212:     int i;
213:     for ( i=0; i<N_PIXEL; i++ ) {
214:         Sbuf[i]=RGBI( IMAX, IMAX, IMAX, 0 );
215:     }
216:     for ( i=0; i<N_PIXEL; i++ ) {
217:         put( 0, i, N_PIXEL-1, i, Sbuf, N_PIXEL*sizeof(short) );

```



```

218: }
219: return ( 0 );
220: }
221:
222: FUNC reverse()
223: {
224:     int i, j;
225:     for ( i=0; i<N_PIXEL; i++ ) {
226:         get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
227:         for ( j=0; j<N_PIXEL; j++ ) {
228:             Sibuf[j] ^= RGBI( IMAX, IMAX, IMAX, 0 );
229:         }
230:         put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
231:     }
232:     return ( 0 );
233: }
234:
235: FUNC maskclear()
236: {
237:     int i, j;
238:     for ( i=0; i<N_PIXEL; i++ ) {
239:         get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
240:         for ( j=0; j<N_PIXEL; j++ ) {
241:             Sibuf[j] ^= RGBI( IMAX, IMAX, IMAX, 0 );
242:         }
243:         put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
244:     }
245:     return ( 0 );
246: }
247: FUNC monotone()
248: {

```

```

249: int i, j;
250: unsigned int s, c;
251: for ( i=0; i<N_PIXEL; i++ ) {
252:     get( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
253:     for ( j=0; j<N_PIXEL; j++ ) {
254:         s=Sibuf[j];
255:         c=(RED(s)*7+GREEN(s)*15+BLUE(s)*28)/256;
256:         Sibuf[j] = RGB( c, c, c );
257:     }
258:     put( 0, i, N_PIXEL-1, i, Sibuf, N_PIXEL*sizeof(short) );
259: }
260: return ( 0 );
261: }

```

リスト7

```

===== main.c =====
1: /****** バラメータ受け渡し用仮変数の実体 *****/
2:
3: unsigned short *par; /* 一時的な引数リスト */
4: unsigned short *ary[10+1]; /* 一時的な配列リスト: X-BASIC の引数は最大 10 個 */
5:
6: /****** コンパイラを通すためのダミー ( 実行されない ) *****/
7:
8: void main()
9: {
10: }

```

リスト8

```

===== pts_curve.c =====
1: /****** Bezier曲線を使った内挿により自由曲線を発生する *****/
2: #include <math.h>
3: #include "act.h"
4: #define MAXPTS1 256 /* 入力点列の長さの最大値 */
5: #define SCALE 32 /* 整数演算の精度を確保するための倍率 */
6: #define MIN_LENGTH (8*OVERSAMPLE*SCALE) /* 再帰分割を打ち切る制定点の間隔 */
7: typedef int ivector[2]; /* 整数値ベクトル */
8: typedef double dvector[2]; /* 実数値ベクトル */
9: ivector P1temp[MAXPTS1][3]; /* テンポラリの制定点 */
10: int Type_pts, N_pts1, N_pts2, Maxpts2;
11: PTS *pts2;
12: #define CPY( V1, V2 ) { V1[0]=V2[0]; V1[1]=V2[1]; } /* ベクトルのコピー */
13: #define SCPY( V1, V2 ) { V1[0]=V2[0]/SCALE; V1[1]=V2[1]/SCALE; } /* 倍率を加味したベクトルのコピー */
14: #define LENGTH( V ) ( sqrt( V[0]*V[0]+V[1]*V[1] ) ) /* ベクトルの長さ */
15: void normalize( v1, v2 ) /* 単位ベクトル化 */
16: {
17:     double l;
18:     l=LENGTH( v1 );
19:     v2[0] = v1[0]/l;
20:     v2[1] = v1[1]/l;
21:     return;
22: }
23:
24: void mult_factor_vec( v1, v2 ) /* 方向は変えないで長さを同じにする */
25: {
26:     double factor;
27:     factor=LENGTH( v2 )/LENGTH( v1 );
28:     v2[0] = factor*v1[0];
29:     v2[1] = factor*v1[1];
30:     return;
31: }
32:
33: void control( p1, p2, p3 ) /* サンプル点から制定点を発生する */
34: {
35:     vector va, vb, vc, norma, normb;
36:     double la, lb;
37:     int i, j, k;
38:     va[0] = (double)(p2[0]-p1[0]); /* 隣のサンプル点への方向ベクトル */
39:     va[1] = (double)(p2[1]-p1[1]);
40:     vb[0] = (double)(p3[0]-p2[0]);
41:     vb[1] = (double)(p3[1]-p2[1]);
42:     norma = (va, norma); /* 単位ベクトル化 */
43:     normb = (vb, normb);
44:     vc[0] = ( norma[0]+normb[0] )/2.0; /* 2等分線をとる */
45:     vc[1] = ( norma[1]+normb[1] )/2.0; /* p2 における接線ベクトル */
46:     mult_factor_vec( vc, va );
47:     mult_factor_vec( vc, vb );
48:     p1[0] = p2[0]-(int)va[0]; /* p2 の隣2つの制定点 */
49:     p1[1] = p2[1]-(int)va[1];
50:     p3[0] = p2[0]+(int)vb[0];
51:     p3[1] = p2[1]+(int)vb[1];
52:     return;
53: }
54:
55:
56: int bezier( s1, s2, s3, s4 ) /* 4つの制定点から Bezier 曲線を発生する */
57: {
58:     ivector s12, s23, s34;
59:     double lx, ly;
60:     #define s1234 s23 /* 再帰が深くなるので制定点を使い回して節約する */
61:     #define s123 s2
62:     #define s234 s3
63:     lx=(double)(s4[0]-s1[0]); /* 制定点間の距離が十分短くなったら */
64:     ly=(double)(s4[1]-s1[1]); /* 再帰を打ち切る */
65:     if ( (lx*lx+ly*ly)<((double)MIN_LENGTH*(double)MIN_LENGTH) ) {
66:         if ( ++N_pts2>Maxpts2 ) return ( 1 );
67:         SCPY( pts2[N_pts2], s2 ); /* 曲線を微小線分で近似する */
68:         if ( ++N_pts2>Maxpts2 ) return ( 1 );
69:         SCPY( pts2[N_pts2], s3 );
70:         if ( ++N_pts2>Maxpts2 ) return ( 1 );
71:         SCPY( pts2[N_pts2], s4 );
72:         return ( 0 );
73:     }
74: }
75:
76: #define MID( V1, V2, V12 ) { V12[0]=(V1[0]+V2[0])/2; V12[1]=(V1[1]+V2[1])/2; }
77: MID( s1, s2, s12 ); /* 中点を取っていく */
78: MID( s2, s3, s23 );
79: MID( s3, s4, s34 );
80: MID( s12, s23, s123 );
81: MID( s23, s34, s234 );
82: MID( s123, s234, s1234 );
83: if ( bezier( s1, s12, s123, s1234 )!=0 ) return ( 1 ); /* 再帰分割 */
84: if ( bezier( s1234, s234, s34, s4 )!=0 ) return ( 1 ); /* 出力点列の不足はエラー */
85:
86: #undef s1234
87: #undef s123
88: #undef s234
89:
90: return ( 0 );

```

```

91:
92:
93:
94: extern unsigned char OVERSAMPLE_NOTYET[];
95: unsigned char CURVE_TOO_MANY[]="入力点の数が多すぎます";
96: unsigned char CURVE_EXHAUSTED[]="出力の配列の大きさが足りません";
97:
98: FUNC pts_curve( dummy ) /* 関数本体 */
99: {
100:     PTS *pts1, int w1, int w2, PTS *pts2;
101:     {
102:         PTS *pts1;
103:         int w1, w2;
104:         int i, j, e, n1, n2, m;
105:         ARGSET( dummy );
106:         ARGSET( 1 );
107:         pts1=PARTOP( 1 );
108:         w1=IVALUE( 2 );
109:         w2=IVALUE( 3 );
110:         ARGSET( 4 );
111:         pts2=PARTOP( 4 );
112:         Maxpts2=1;
113:         for ( i=0; i<DIM( 4 ); i++ ) {
114:             Maxpts2 += ( SUFFIX( 4, i+1 )+1 );
115:         }
116:         Maxpts2 /= PTSSIZE; /* 出力点列の長さの最大値 */
117:
118:         N_pts1=pts1[0][0]; /* 入力点列の長さ */
119:         Type_pts=pts1[0][1]; /* 入力点列のタイプ */
120:         if ( N_pts1>MAXPTS1 ) {
121:             asm ( " lea.l _CURVE_TOO_MANY,a1" );
122:             #endif
123:         }
124:         #asm
125:         lea.l _CURVE_TOO_MANY,a1
126:         #endasm
127:         #endif
128:         return ( 1 );
129:     }
130:     if ( pts1[0][2]>OVERSAMPLE ) {
131:         #ifdef _GNUC
132:             asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
133:         #else
134:             #asm
135:             lea.l _OVERSAMPLE_NOTYET,a1
136:             #endasm
137:         #endif
138:         return ( 1 );
139:     }
140:     for ( i=1; i<N_pts1; i++ ) {
141:         P1temp[i][0][0]=pts1[i][0]*SCALE; /* サンプル点を制定点 */
142:         P1temp[i][0][1]=pts1[i][1]*SCALE;
143:     }
144:     /* 制定点の前処理 ... サンプル点を3等分する */
145:     for ( i=1; i<N_pts1; i++ ) {
146:         P1temp[i][1][0]=(P1temp[i][0][0]*2+P1temp[i+1][0][0])/3;
147:         P1temp[i][1][1]=(P1temp[i][0][1]*2+P1temp[i+1][0][1])/3;
148:         P1temp[i][2][0]=(P1temp[i][0][0]+P1temp[i+1][0][0]*2)/3;
149:         P1temp[i][2][1]=(P1temp[i][0][1]+P1temp[i+1][0][1]*2)/3;
150:     }
151:     if ( Type_pts==CYCLIC ) {
152:         P1temp[N_pts1][1][0]=(P1temp[N_pts1][0][0]*2+P1temp[1][0][0])/3;
153:         P1temp[N_pts1][1][1]=(P1temp[N_pts1][0][1]*2+P1temp[1][0][1])/3;
154:         P1temp[N_pts1][2][0]=(P1temp[N_pts1][0][0]+P1temp[1][0][0]*2)/3;
155:         P1temp[N_pts1][2][1]=(P1temp[N_pts1][0][1]+P1temp[1][0][1]*2)/3;
156:     }
157:     /* 制定点の発生 */
158:     for ( i=2; i<N_pts1; i++ ) {
159:         control( P1temp[i-1][2], P1temp[i][0], P1temp[i][1] );
160:     }
161:     if ( Type_pts==CYCLIC ) {
162:         control( P1temp[N_pts1-1][2], P1temp[N_pts1][0], P1temp[N_pts1][1] );
163:         control( P1temp[N_pts1][2], P1temp[1][0], P1temp[1][1] );
164:     }
165:     /* 自由曲線の生成 */
166:     SCPY( pts2[1], P1temp[1][0] ); /* 始点はマニュアルでコピー */
167:     if ( w1<=0 || w2<=0 ) {
168:         m=1; /* 線の太さを点列レベルで ( 頂点ごとに ) 指定されている */
169:     } else {
170:         m=0; /* 線の太さはコマンドレベル ( 点列全体 ) で指定されている */
171:     }
172:     N_pts2=1;
173:     for ( i=1; i<N_pts1; i++ ) {
174:         n1=N_pts2;
175:         e=bezier( P1temp[i][0], P1temp[i][1], P1temp[i][2], P1temp[i+1][0] );
176:         n2=N_pts2;
177:         if ( m ) {
178:             for ( j=n1; j<n2; j++ ) {
179:                 pts2[j][2]=(pts1[i][2]*(n2-j)+pts1[i+1][2]*(j-n1))/(n2-n1);
180:             }
181:         }
182:     }

```



```

183: if ( e==0 && Type_pts==CYCLIC ) { /* 点列が循環する場合 */
184:     n1=N_pts2;
185:     e=bezier( Ptemp[N_pts1][0], Ptemp[N_pts1][1], Ptemp[N_pts1][2], Ptemp[1][0] );
186:     n2=N_pts2;
187:     N_pts2--; /* 終点は始点と一致するので減らす */
188:     if ( m ) {
189:         for ( j=n1; j<n2; j++ ) {
190:             pts2[j][2]=(pts1[j][2]*(n2-j)+pts1[j+1][2]*(j-n1))/(n2-n1);
191:         }
192:     }
193: }
194: if ( e!=0 ) {
195: #ifdef _GNUC_
196:     asm ( " lea.l _CURVE_EXHAUSTED,a1" );
197: #else
198:     #asm

```

```

199:     lea.l _CURVE_EXHAUSTED,a1
200: #endasm
201: #endif
202:     return ( 1 );
203: }
204: pts2[0][0]=N_pts2; /* ヘッダをつける */
205: pts2[0][1]=Type_pts;
206: pts2[0][2]=OVERSAMPLE;
207: if ( m==0 ) {
208:     for ( i=1; i<N_pts2; i++ ) { /* 各区間の幅を補間しながら設定する */
209:         pts2[i][2]=(w1*(N_pts2-i)+w2*(i-1))/(N_pts2-1);
210:     }
211: }
212: return ( 0 );
213: }

```

リスト9

```

===== pts_procs.c =====
1: /****** 点列の移動および接続 *****/
2:
3: #include "anti.h"
4:
5: unsigned char OVERSAMPLE_NOTYET[]="オーバーサンプリング座標に変換してください";
6:
7: unsigned char MOVE_INCOMPATIBLE[]="移動先の点列とサイズが合いません";
8:
9: FUNC pts_move( dummy )
10: DUMMY dummy;
11: /* PTS *pts1, int x, int y, PTS *pts2 */
12: {
13:     PTS *pts1, *pts2;
14:     int x, y, i;
15:     int n1, n2;
16:
17:     ARGSET( dummy );
18:     ARYSET(1);
19:     pts1=PARTOP(1);
20:     x=IVALUE(2);
21:     y=IVALUE(3);
22:     ARYSET(4);
23:     pts2=PARTOP(4);
24:     n2=1;
25:     for ( i=0; i<DIM(4); i++ ) {
26:         n2 += ( SUFFIX(4,i+1)+1 );
27:     }
28:     n2 /= PTSSIZE;
29:     if ( pts1[0][2]!=OVERSAMPLE ) {
30: #ifdef _GNUC_
31:         asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
32: #else
33:         #asm
34:         lea.l _OVERSAMPLE_NOTYET,a1
35: #endasm
36: #endif
37:         return ( 1 );
38:     }
39:     n1=pts1[0][0];
40:     if ( (n1+1)>n2 ) {
41: #ifdef _GNUC_
42:         asm ( " lea.l _MOVE_INCOMPATIBLE,a1" );
43: #else
44:         #asm
45:         lea.l _MOVE_INCOMPATIBLE,a1
46: #endasm
47: #endif
48:         return ( 1 );
49:     }
50:     pts2[0][0]=n1;
51:     pts2[0][1]=pts1[0][1];
52:     pts2[0][2]=OVERSAMPLE;
53:     for ( i=1; i<n1; i++ ) {
54:         pts2[i][0]=pts1[i][0]*x;
55:         pts2[i][1]=pts1[i][1]*y;
56:         pts2[i][2]=pts1[i][2];
57:     }
58:     return ( 0 );
59: }
60:
61: unsigned char APPEND_INSUFFICIENT[]="移動先の点列のサイズが足りません";
62:
63: FUNC pts_append( dummy )
64: DUMMY dummy;
65: /* PTS *pts1, int x, int y, PTS *pts2 */
66: {
67:     PTS *pts1, *pts2;
68:     int x, y, i;
69:     int n1, n2;
70:
71:     ARGSET( dummy );
72:     ARYSET(1);
73:     pts1=PARTOP(1);

```

```

74:     ARYSET(2);
75:     pts2=PARTOP(2);
76:     if ( pts1[0][2]!=OVERSAMPLE || pts2[0][2]!=OVERSAMPLE ) {
77: #ifdef _GNUC_
78:         asm ( " lea.l _OVERSAMPLE_NOTYET,a1" );
79: #else
80:         #asm
81:         lea.l _OVERSAMPLE_NOTYET,a1
82: #endasm
83: #endif
84:         return ( 1 );
85:     }
86:     n1=1;
87:     for ( i=0; i<DIM(1); i++ ) {
88:         n1 += ( SUFFIX(1,i+1)+1 );
89:     }
90:     n1 /= PTSSIZE;
91:     n2=pts2[0][0];
92:     if ( n1<(pts1[0][0]+n2) ) {
93: #ifdef _GNUC_
94:         asm ( " lea.l _APPEND_INSUFFICIENT,a1" );
95: #else
96:         #asm
97:         lea.l _APPEND_INSUFFICIENT,a1
98: #endasm
99: #endif
100:         return ( 1 );
101:     }
102:     n1=pts1[0][0];
103:     pts1[0][0]=n1+n2-1;
104:     x=pts1[n1][0]-pts2[1][0]; /*pts1 の終点とpts2の始点を一致させる*/
105:     y=pts1[n1][1]-pts2[1][1];
106:     for ( i=1; i<n2; i++ ) {
107:         pts1[i+n1][0]=pts2[i+1][0]*x;
108:         pts1[i+n1][1]=pts2[i+1][1]*y;
109:         pts1[i+n1][2]=pts2[i+1][2];
110:     }
111:     return ( 0 );
112: }
113:
114: unsigned char OVERSAMPLE_ALREADY[]="オーバーサンプリング済みです";
115:
116: FUNC pts_oversample( dummy )
117: DUMMY dummy;
118: /* PTS *pts */
119: {
120:     PTS *pts;
121:     int n, i;
122:
123:     ARGSET( dummy );
124:     ARYSET(1);
125:     pts=PARTOP(1);
126:     if ( pts[0][2]!=OVERSAMPLE ) {
127: #ifdef _GNUC_
128:         asm ( " lea.l _OVERSAMPLE_ALREADY,a1" );
129: #else
130:         #asm
131:         lea.l _OVERSAMPLE_ALREADY,a1
132: #endasm
133: #endif
134:         return ( 1 );
135:     }
136:     pts[0][2]=OVERSAMPLE;
137:     n=pts[0][0];
138:     for ( i=1; i<n; i++ ) {
139:         pts[i][0] = OVER( pts[i][0] );
140:         pts[i][1] = OVER( pts[i][1] );
141:     }
142:     return ( 0 );
143: }

```

リスト10

```

===== anti.h =====
1: /****** 汎用マクロなどの定義 *****/
2:
3: #define PTSSIZE 3 /* 輪郭を点列で表現する */
4: typedef int PTS[ PTSSIZE ];
5:
6: #define N_PIXEL 512 /* スクリーンのサイズは 512x512 ピクセル */
7:
8: #define OVERSAMPLE 8 /* オーバーサンプリング回数 */
9: #define OVER2 (OVERSAMPLE*OVERSAMPLE) /* 1ピクセルあたりのサブピクセル数*/
10:
11: /* 通常の座標からオーバーサンプリング座標に変換する。など */
12: #define OVER( X ) ((X)*OVERSAMPLE+(OVERSAMPLE/2))
13: #define PIX( X ) ((X)/OVERSAMPLE)
14: #define SUBPIX( X ) ((X)/OVERSAMPLE)
15:
16: /* 点列のフォーマット *****/
17:
18: PTS *pts; /* 宣言してあるとする */
19:
20: pts[0][0] 点列を構成する点の数
21: pts[0][1] 点列のタイプ (片道通行か循環しているか)
22: pts[0][2] オーバーサンプリング回数 (ここが OVERSAMPLE でないなら描画回数はエラーになる)
23:
24: ** 第 i 点の情報 ( 1 ≤ i ≤ pts[0][0] ) **
25:
26: pts[i][0] x 座標

```

```

27: pts[i][1] y 座標
28: pts[i][2] 線の幅 (この値が OVERSAMPLE なら 1ピクセルぶんの幅)
29:
30: *****
31:
32: /* 点列のタイプ:片道通行か循環しているのか ... pts[0][1] の値 */
33:
34: #define NORMAL 0
35: #define CYCLIC 1
36:
37:
38: /* X-BASIC からの引数をアクセスする */
39:
40: typedef int FUNC; /* X-BASIC の外部関数: 戻り値はエラーコード */
41: typedef int DUMMY; /* Cとの引数の受け渡しの違いを吸収するダミー引数 */
42:
43: extern unsigned short *par; /* 一時的な引数リスト */
44: extern unsigned short *ary[10+1]; /* 一時的な配列リスト: X-BASIC の引数は最大 10 個 */
45:
46: #define ARGSET( A ) { par=(unsigned short *)(&A); } /*引数*/
47: #define ATOP( I ) ( (I)*5-4 ) /*第 I 引数の先頭*/
48: #define TYPE( I ) ( par[ATOP(I)] ) /*引数の型 */
49: #define IVALUE( I ) ( par[ATOP(I)+3]<<16 | par[ATOP(I)+4] ) /*intの値*/
50: #define CVALUE( I ) ( par[ATOP(I)+4] ) /*charの値*/
51: #define ARYSET( I ) { ary[I]=(unsigned short *)IVALUE(I); } /*配列 */
52: #define DIM( I ) ( ary[I][2]+1 ) /*配列の次元 */
53: #define ELEMENT( I ) ( ary[I][3] ) /*配列要素のサイズ*/

```



```

54: #define SUFFIX( I, J ) ( ary[I][J+3] ) /*第 J 添字の最大値*/
55: #define ARYTOP( I ) ( &(ary[I][DIM(I)+3+2]) ) /*配列の先頭*/
56: #define IARYTOP( I ) ( (int *)ARYTOP(I) ) /*int 配列の先頭*/
57: #define CARYTOP( I ) ( (unsigned char *)ARYTOP(I) ) /*char 配列の先頭*/
58: #define PARYTOP( I ) ( (PTS *)ARYTOP(I) ) /*PTS 配列の先頭*/
59:
60: /* その他、便利なマクロ */
61:
62: #define ABS( X ) (((X)>0)?(X):(-(X))) /* X の絶対値 */
63: #define SGN( X ) (((X)>0)?1:(((X)<0)?(-1):0)) /* X の符号 (正負または零) */
64:
65: #define MIN( X, Y ) (((X)>(Y))?Y:(X)) /* X,Y のうち小さいほう */
66: #define MAX( X, Y ) (((X)>(Y))?X:(Y)) /* X,Y のうち大きいほう */
67:
68: /* アンチエイリアシング関係 ... ペイントおよびソリッドスキャンコンバージョン */
69:
70: #define N_TILE 8 /* 格納できるタイルパターンの数 */
71: #define N_TONE 8 /* 格納できるトーンパターンの数 */
72: #define T_SIZE 64 /* タイル及びトーンパターンの大きさ */
73:
74: #define COLOR 0 /* 単色で塗り潰す */
75: #define TILE 1 /* タイルパターンにしたがって色をつける */
76:
77: #define OFF_NTP 0 /* トーンは使わない・ベタ塗り */
78: #define ON_VTP 1 /* トーンを使う・ベタ塗り */
79: #define OFF_TP 2 /* トーンは使わない・下地は透ける */
80: #define ON_TP 3 /* トーンを使う・下地は透ける */
81: #define ON 1 /* トーンを使う */
82: #define TP 2 /* 下地は透ける */

```

```

83:
84: /* R,G,B ごとの輝度を得るためのマスクとビットシフト */
85: #define VMASK_B 62 /* 0b000000000000000111110 */
86: #define VMASK_R 1984 /* 0b0000011111000000000 */
87: #define VMASK_G 63488 /* 0b11111000000000000000 */
88:
89: #define SHIFT_B 1
90: #define SHIFT_R 6
91: #define SHIFT_G 11
92:
93: /* 輝度ビットの値を取り出すためのマスク */
94: #define VMASK_I 1 /* 0b00000000000000000001 */
95: #define VMASK_VMASK_R /* 輝度の代表値は赤ブレンから取ってくる */
96: #define VSHIFT SHIFT_R
97: #define FMASK VMASK_I /* ペイント済みフラグには輝度ビットを用いる */
98: #define SIMASK (FMASK|VMASK)
99:
100: /* R,G,B および R,G,B,I からカラーコードを計算する */
101: #define RGB(R,G,B) ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G)
102: #define RGBI(R,G,B,I) ((B)<<SHIFT_B|(R)<<SHIFT_R|(G)<<SHIFT_G|(I))
103:
104: /* カラーコードから R,G,B,I 成分を取り出す */
105: #define BLUE(C) ((C)&VMASK_B)>>SHIFT_B
106: #define RED(C) ((C)&VMASK_R)>>SHIFT_R
107: #define GREEN(C) ((C)&VMASK_G)>>SHIFT_G
108: #define INTENSITY(C) ((C)&VMASK_I)
109: #define VALUE(C) ((C)&VMASK)>>VSHIFT
110: #define IMAX 31 /* R,G,B の輝度の最大値 */

```

リスト11

```

===== anti.s =====
1: ***** 外部関数ヘッダ *****
2: # pts_curve.c
3: # pts_curve( PTS *pts1, int w1, int w2, PTS *pts2 )
4: # pts_procs.c
5: # pts_append( PTS *pts1, PTS *pts2 )
6: # pts_move( PTS *pts1, int x, int y, PTS *pts2 )
7: # pts_oversample( PTS *pts )
8: # aa_lines.c
9: # aa_lines( PTS *pts, int c )
10: # lines( PTS *pts, int c )
11: # aa_scanconv.c
12: # aa_scanconv( PTS *pts, int cmode, int c/n_tile, int tmode, int n_tone )
13: # scanconv( PTS *pts, int c )
14: # aa_paint.c
15: # aa_paint( int x, int y, int cmode, int c/n_tile, int tmode, int n_tone )
16: # aa_procs.c
17: # tile_get( int n, int x1, int y1, int x2, int y2 )
18: # tone_get( int n, int x1, int y1, int x2, int y2 )
19: # whitepaper( void )
20: # reverse( void )
21: # maskclear( void )
22: # monotone( void )
23: # インフォメーション・テーブル
24: do.l X_INIT
25: do.l X_RUN
26: do.l X_END
27: do.l X_SYS
28: do.l X_BRK
29: do.l X_CTRL_D
30: do.l X_RES1
31: do.l X_RES2
32: do.l PTR_TOKEN
33: do.l PTR_PARAM
34: do.l PTR_EXEC
35: do.l 0,0,0,0,0
36: X_INIT:
37: X_RUN:
38: X_END:
39: X_SYS:
40: X_BRK:
41: X_CTRL_D:
42: X_RES1:
43: X_RES2:
44: rts
45:
46: # 関数名テーブル
47: PTR_TOKEN:
48: do.b 'pts_curve',0
49: do.b 'pts_append',0
50: do.b 'pts_move',0
51: do.b 'pts_oversample',0
52: do.b 'lines',0
53: do.b 'aa_lines',0
54: do.b 'scanconv',0
55: do.b 'aa_scanconv',0
56: do.b 'aa_paint',0
57: do.b 'tile_get',0
58: do.b 'tone_get',0
59: do.b 'whitepaper',0
60: do.b 'reverse',0
61: do.b 'maskclear',0
62: do.b 'monotone',0
63: do.b 0
64: .even
65:
66: # パラメータ・テーブルへのポインタ
67:
68: PTR_PARAM:
69: do.l PTS_CURVE_PAR
70: do.l PTS_APPEND_PAR
71: do.l PTS_MOVE_PAR
72: do.l PTS_OVERSAMPLE_PAR
73: do.l LINES_PAR
74: do.l AA_LINES_PAR
75: do.l SCANCONV_PAR
76: do.l AA_SCANCONV_PAR
77: do.l AA_PAINT_PAR
78: do.l TILE_GET_PAR
79: do.l TONE_GET_PAR
80: do.l WHITEPAPER_PAR
81: do.l REVERSE_PAR
82: do.l MASKCLEAR_PAR
83: do.l MONOTONE_PAR
84:
85: # パラメータ・テーブル
86:
87: int_val: equ $0002 /* int
88: pts_ary: equ $0052 /* 1D-array of PTS ( 2D-array of int )
89: rlc_ary: equ $0037 /* 1D-array of float,int,char
90: void_ret: equ $ffff /* void
91:
92: PTS_CURVE_PAR:

```

```

93: do.w PTS_ary
94: do.w int_val
95: do.w int_val
96: do.w PTS_ary
97: do.w void_ret
98: PTS_APPEND_PAR:
99: do.w PTS_ary
100: do.w PTS_ary
101: do.w void_ret
102: PTS_MOVE_PAR:
103: do.w PTS_ary
104: do.w int_val
105: do.w int_val
106: do.w PTS_ary
107: do.w void_ret
108: PTS_OVERSAMPLE_PAR:
109: do.w PTS_ary
110: do.w void_ret
111: LINES_PAR:
112: do.w PTS_ary
113: do.w int_val
114: do.w void_ret
115: AA_LINES_PAR:
116: do.w PTS_ary
117: do.w int_val
118: do.w void_ret
119: SCANCONV_PAR:
120: do.w PTS_ary
121: do.w int_val
122: do.w void_ret
123: AA_SCANCONV_PAR:
124: do.w PTS_ary
125: do.w int_val
126: do.w int_val
127: do.w int_val
128: do.w int_val
129: do.w void_ret
130: AA_PAINT_PAR:
131: do.w int_val
132: do.w int_val
133: do.w int_val
134: do.w int_val
135: do.w int_val
136: do.w int_val
137: do.w void_ret
138: TILE_GET_PAR:
139: do.w int_val
140: do.w int_val
141: do.w int_val
142: do.w int_val
143: do.w int_val
144: do.w void_ret
145: TONE_GET_PAR:
146: do.w int_val
147: do.w int_val
148: do.w int_val
149: do.w int_val
150: do.w int_val
151: do.w void_ret
152: WHITEPAPER_PAR:
153: do.w void_ret
154: REVERSE_PAR:
155: do.w void_ret
156: MASKCLEAR_PAR:
157: do.w void_ret
158: MONOTONE_PAR:
159: do.w void_ret
160:
161: # 関数へのポインタ
162: PTR_EXEC:
163: do.l pts_curve
164: do.l pts_append
165: do.l pts_move
166: do.l pts_oversample
167: do.l lines
168: do.l aa_lines
169: do.l scanconv
170: do.l aa_scanconv
171: do.l aa_paint
172: do.l tile_get
173: do.l tone_get
174: do.l whitepaper
175: do.l reverse
176: do.l maskclear
177: do.l monotone
178: .even

```

▶ 7月号を買いに行った。どの本屋にもなかった。はるかかなたの本屋まで行ってやっと買った。急いで帰って読んだ。落丁していた。その本屋までとりかえに行った。……悲しいぞ。

板垣 央(16)千葉県

X-BASICによる画像処理

後処理によるジャギーの除去

Nakano Shuichi 中野 修一

X68000によるグラフィックの扱い

ふつうコンピュータグラフィックというのは、画面上の点の色の集まりに還元される。さらに、テレビなどでは色の基本は緑赤青で作られる。これら光の3原色でだいたいの色は作れるわけだ。

緑+赤=黄

緑+青=水色(シアン)

赤+青=紫(マゼンタ)

緑+赤+青=白

のような具合だ。

X68000では16色、256色、65536色のグラフィック画面を扱える。16は2の4乗、256は2の8乗、65536は2の16乗となる。これらはコンピュータにとってはものすごくきりのいい数字だから、処理も速いしメモリ効率もいい。

これらに対して、標準パレットでは、GRBI

表1

```
key 1,"files @@"
key 2,"load @M"
key 3,"auto "
key 4,"list @M"
key 5,"run @M"
key 6,"/*"
key 7,"width "
key 8,"end"
key 9,"func "
key 10,"system"
key 11,"chdir @@"
key 12,"chdrv @@"
key 13,""
key 14,""
key 15,""
key 16,"sa.@@test@M!gbc test||test@M"
key 17,"sa.@@test@M!ed test.bas@M"
key 18,"img_l@Aoad(@A@I@I@I@I@I)@M"
key 19,""
key 20,""
```

図1



3×3ドットのエリアで見たとき、境界が直線に並んでいると思われる場合は中心のドットは修正しないほうがいい

GGRRRBBB

GGGGGRRRRRRBBBBBI

というふうに2進数の各桁が対応している。Gは緑,Rは赤,Bは青,Iがつくとその色が明るくなると思っておけばいい。ついていると1、消えていると0の値をとる。

16色の場合を考えよう。4(2進数で0100という数値)はGRBIのRがついた状態とみなされる。これは暗い赤に相当する。赤と緑を混ぜた明るい黄色なら13(1101)というふうになる。

256色の場合も同様に数値を2進数で表したときの各桁の状況が色の成分を決めている。ただ、256色のときは暗い緑と倍明るい緑があったり、暗い赤、倍明るい赤とその倍明るい赤、暗い青、倍明るい青とその倍明るい青のようになっているだけだ。256色モードでは赤と青を3段階(8階調)、緑だけ2段階(4階調)で表すことになっている。

65536色は緑赤青各32階調に明るさが1段階加わったものだ。

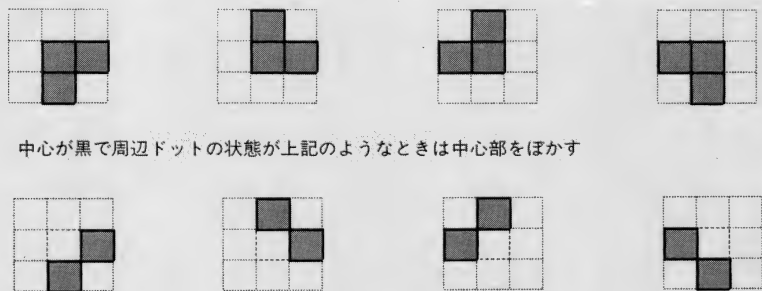
要するに色は数字で扱われる。ある数値がどんな色になるかは2進数で表せばわかる。試しに45627という数値を色にしたとき、赤成分はどのようになっているかを見よう。BASICから、

?bin\$(45627)

とすると、

101100100011101

図2



中心が黒で周辺ドットの状態が上記のようなときは中心部をぼかす

中心が黒以外で周辺ドットが上記のようなときは中心部を軽くぼかす

すでに描かれた絵のギザギザした部分を滑らかにする、そんな処理はできないでしょうか(もちろん、ほかしや手作業じゃなく)。ここでは3通りのアプローチで輪郭線を綺麗にすることを考えてみます。同時にX-BASICでのグラフィック処理の基本から見えていきましょう。

と答えが出る。下7~11桁の5桁が赤成分だから、01000=16となる。

このように色をRGB成分に分離して操作することがグラフィック処理の基本となる。

ジャギーをなくす

今回はすでに描いた絵からジャギーを消す、という処理を考えてみたい。

情報量が少ないので完全な処理は理論的に不可能だ。また、ちゃんとした補間をやるととても重いので、以下は補間といっても平均をとっているだけと考えていい。

これをX-BASICで記述するわけだが、処理自体はともかく、今回のプログラムは高速化などはほとんど考えられていないので、内容的にBASICインタプリタ上で動かすのは相当無理がある。処理範囲を狭くして動作チェックを行うのが関の山といところだろう。

さらにいえば、動作チェックもコンパイラからのほうがいい。これならエディタからコンパイラを起動しても変わらないような気がするが、BASICのプログラムには行番号が必要なのに、ED.Xを始めあらゆるエディタにはリナンバー機能がついていない。よってBASICから作業を行うのがもっともよいことになる。

プログラムを直すごとにBASICを抜けてコンパイラを起動するのは面倒なのでチ

チャイルドプロセスを使う。さらに、いちいちチャイルドプロセスを起動してコンパイラにたくさんのオプションを与えるのは面倒なので、コンパイラの起動はバッチファイル、BASICからはファンクションキー1発でコンパイル実行できるようにするとよい。

表1のようなファンクションキー設定だとシフト+F6キーで即座にコンパイル実行できる。RUNコマンドの代わりと思えばいい(メモリの少ない人はできません)。

輪郭パターンでの補正

まず2月号で行った局所補間つき画面拡大プログラムを見てみよう。これは256×256ドットの絵を512×512ドットに拡大するものだ。ドットをそのまま大きくすると当然モザイクになる。かといって単純に周辺の色と補間して拡大するとボケボケの絵になる。これを防ぐため、輪郭部分を保護しつつ、全体にぼかしをかけることになった(ただし手抜き処理なので斜め方向は見えない)。

今回のアンチエイリアシング(正確には違うが)でもぼかしを使うことを考えてみよう。絵の輪郭を抽出することは容易だが、そこからベクトルを得ることはちょっと難しいので本格的な処理は私にはできない。

2月号では取り込み画像を対象にしていたため、輪郭保護に重点をおいて明度変化



元画像 (協力: 高橋哲史)



変換後

の激しい部分はほっという、それ以外の部分にぼかしをかけていた。今度はこれとは逆に、明度変化の緩やかな部分は元絵を残し、明度差の激しい部分を選択的にぼかすことになる。しかし、なんでもかんでもぼかすと元絵を大きく損なうので、ぼかさなくてもいい場合を考えよう。

中心が黒でかつ、上下や左右にも黒い点が連続するときはぼかす必要はない(図1)。あまり考えずにアンチエイリアシングをやったよさそうなのは、図2に示されるパターンだ、としよう。

まず、輪郭線部分を取り出し、その周辺の状況(輪郭が連続しているかどうか)を配列に読み込む。ある点の周りには8つの点が存在するので、これをビットごとにchar型配列に入れる。

すると256とおりの場合分けができるので、一気にswitchで最適な処理をすると



拡大するとこうなる

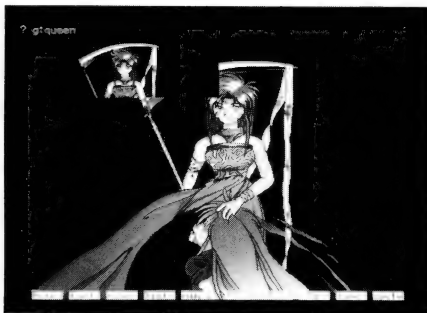
いうのもいいんだが、ここでは最小限の処理にとどめておく。拡張はご自由に。

中心点が黒かどうかで図2の上下の処理を選択し、ほぼ全ドットに渡って置き換えを実行する。ぼかしは上下左右のドットの色をRGBごとに重みつきで平均することで行っている。点ごとにだぶった処理を行っているがとりあえず気にしない。これでかなりジャギーが減ったはずだ。

リスト1

```
10 /* ----- initialize ----- */
20 screen 1,3,1,1
30 str nam
40 int g_dat(4,2),col,d(4,2),c(4)
50 int blue=0,red=1,green=2,i,q=3333
60 char fl(511,511),fl2(511,511)
70 /* ----- main ----- */
80 input nam
90 pic_load(nam+".pic",0,0)
100 edge():beep
110 jag():beep
120 bokasi()
130 input i
140 end
150 /* ----- */
160 func edge()
170 for y=1 to 510
180   for x=1 to 510
190     c(0)=point(x, y)
200     if c(0)=0 then {
210       fl2(x,y)=1:/*pset(x,y,1)
220       /* c(1)=point(x+1,y) :/* 2 エッジ検出部の名残
230       /* c(2)=point(x,y-1) :/* 4 0 1
240       /* c(3)=point(x,y+1) :/* 3
250       /* c(4)=point(x-1,y) :/*
260       /* if c(1)>1 or c(2)>1 or c(3)>1 or c(4)>1 then {
270       /*   fl(x,y)=1
280       /* }
290       /* } else fl2(x,y)=0
300     next
310 next
320 endfunc
330 /* ----- */
340 func jag()
350 for y=1 to 510
360   for x=1 to 510
370     col=0
380     /* if fl2(x-1,y-1)=1 then col=col+128
390     /* if fl2(x,y-1)=1 then col=col+64
400     /* if fl2(x+1,y-1)=1 then col=col+32 :/* 7 6 5
410     /* if fl2(x-1,y)=1 then col=col+16 :/* 4 3
420     /* if fl2(x+1,y)=1 then col=col+8 :/* 2 1 0
430     /* if fl2(x-1,y+1)=1 then col=col+4
440     /* if fl2(x,y+1)=1 then col=col+2
450     /* if fl2(x+1,y+1)=1 then col=col+1
460     fl(x,y)=col
```

```
460 next
470 next
480 endfunc
485 /* ----- */
490 func bokasi()
500 for y=1 to 510 :/* ぼかし処理
510   for x=1 to 510
520     if fl2(x,y)=1 then {
530       if (fl(x,y) and &B10010) >16 then fuz(x,y,0)
540       if (fl(x,y) and &B1010) >8 then fuz(x,y,0)
550       if (fl(x,y) and &B1001000)>64 then fuz(x,y,0)
560       if (fl(x,y) and &B10100000)>64 then fuz(x,y,0)
570     } else {
580       if (fl(x,y) and &B1010000)>64 then fuz(x,y,4)
590       if (fl(x,y) and &B1001000)>64 then fuz(x,y,4)
600       if (fl(x,y) and &B1010) >8 then fuz(x,y,4)
610       if (fl(x,y) and &B10010) >16 then fuz(x,y,4)
620     }
630   next
640 next
650 endfunc
660 /* ----- */
670 func fuz(x,y,p)
680 c(0)=point(x, y)
690 if c(0)>1 then {
700   c(1)=point(x,y+1)
710   c(2)=point(x+1,y)
720   c(3)=point(x-1,y)
730   c(4)=point(x,y-1)
740   get_rgb()
750   for m=blue to green
760     d(i,m)=(g_dat(0,m)*p+g_dat(1,m)+g_dat(2,m)+g_dat(3,m)+
770       g_dat(4,m))*p/4)
780   next
790   pset(x, y, rgb(d(0,red),d(0,green),d(0,blue)))
800 }
810 endfunc
820 /* ----- */
830 func get_rgb()
840 for m=blue to green
850   for i=0 to 4
860     g_dat(1,m)=(c(1) mod (1 shl(5*m+6)))shr (m*5+1)
870   next
880 next
890 endfunc
900 next
910 endfunc
920 /* ----- */
```

縮小中

ただし、ばかしを直接画面に描いているので、画面処理されたあとのデータを対象に処理が進んでしまう。これはふつうダサイやり方と呼ばれる。スキャンラインごとに処理をすることもできるので、小さなバッファを取って影響がなくなってから書き込むというのが正しいのだろう。多少処理が複雑になることと、モノがばかしだけに周りに影響が出て問題ないんじゃないかという楽観論からこのままにしておいた。

本当は画面分バッファを取って、
int gbuff1(511, 511)
のようにしたかったのだが、こういった配列を2つ取ると多くの人のメモリでは収まらないはずなのであきらめた。

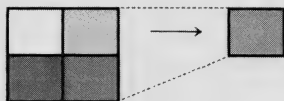
それでも512Kバイト分の配列を取っているので、BASIC.CNFを変更してフリーエリアを広げておいてほしい。あとは、PIC.FNC(1990年6月号)をお持ちの方はそのまま、ない方は“pic_”を“img_”に変更して使えばいい。

なお、PIC.FNCを使ったプログラムをコンパイルする場合、ほかのヘッダファイルなどをいじらない限り、パラメータを省略することはできないので注意しよう。ロード先頭座標やセーブする範囲はその都度指定する。当然コンパイル時にはPICLIB.Aも指定すること。

256ドット縮小

さて、アンチエイリアシングは悪く言えば不十分なドット数をごまかす手法だ。ふつうはオーバーサンプリングといって「たくさんドットがある」つもりで計算しておき、「実は少なかったんだ」といって1ドット

図3



4点の輝度の構成を平均して新しい輝度を得る

に詰め込むときに平均を取ってやる。

X68000の512×512ドットというのは十分なようで実は少ないともいえる。32ビットマシンなら1024×1024以上が標準だろうし、グラフィックのジャギーを見るとこれくらいはほしくなる。しかし、現状のツールでは真っ正直な線しか考えてない。しかたないからアンチエイリアシングするわけだが、現状の512×512ドットのモードがすでにオーバーサンプリングされているとみなすのだろうか？

実用上必要なのは綺麗な絵であって高い解像度ではない。なにかとかさむ高解像度の絵より256×256ドットの絵が好まれる場合もある。当然、解像度が低いとジャギーが目立つわけだ。256ドット以下のワンポイント的に使われる絵だって綺麗なほうがいいに決まっている。

となると話は簡単。512×512ドットモードで（当たり前のグラフィックツールを使って）描いた絵を縮小してやればいい。手抜きのグラフィックツールを使うとドットを間引かれるので、ここではプログラムによって平均化された画像を作ることにしよう。

図3のようになった4点をRGB別にして平均し色を決める。小さな画面ならわざわざファイルに書き出したり、大きなバッファを取らなくても画面にそのまま表示できるので結果はリスト2のように単純だ。

ここでは画面の初期化やファイルのロード/セーブを行っていない。BASICで実行しても耐えられない速度ではないということが理由だが、コンパイルして実行したほうがいいに決まっている。必要な人は各自で対応してほしい。また、ファイルのロード時にわざわざinputを使うのも面倒だという場合はコマンドラインから文字を取り込むようにするとよい。Cユーザーズマニュアル参照のこと。

1/4補正つき拡大

なにも画像を小さくしなくても、疑似的にオーバーサンプリングできるようにする手もある。簡単にいえば昔使った4倍拡大アルゴリズムで拡大しておいて、今度はそれをモザイク化して1/4の画像を作り出す、という手だ。拡大時に輪郭補正と周りとの平均化を行うので、不正確ながら高解像度のデータを合成することができよう。

あとは通常のアンチエイリアシングと同様に面積比（といっても4つの平均だが）で色を決定すればいいわけだ。

今回は輪郭色を黒のみに限定して黒のみの補間を行うことにする。それ以外の色ではなにもしない。理由はすぐに縮めるんだからなにもしなくても変わらないからと、黒を残しておけば最初に作ったプログラムをそのまま使ってさらにアンチエイリアシングを図ることもできるからだ。

こうしてできたプログラムがリスト3。画面上の256×256の部分512×512のエリアに拡大する。あらかじめ512×512ドットの絵を1/4ずつに分けてセーブしておいてほしい。

プログラムは同じ画面でもかちあわないように画像の右下から順に処理を進めていく。まず基準点の色を拡大された部分の右隅に打ち、上、左、左上の各ドットの内容から残りの3点の状況を決定する。「両方とも黒ならあいだも黒」というのが基本コンセプトだ。

これだと、左斜めは検出するが、逆の斜めは検出できないので逆斜め専用のループも入れてある。

基準点黒以外ならなにもしないのでその色を4点に置く。このあたりは改良の余地があるかもしれない。

輪郭を黒に限定しない場合なら、単に画

リスト2

```
10 int c(3)
20 int blue=0,red=1,green=2
30 int g_dat(3,2),r,g,b
60 for y=0 to 255
70   for x=0 to 255
80     c(0)=point(x*2,y*2)
90     c(1)=point(x*2+1,y*2)
100    c(2)=point(x*2,y*2+1)
120    c(3)=point(x*2+1,y*2+1)
210    for m=blue to green
220      for l=0 to 3
230        g_dat(l,m)=(c(1) mod (1 shl(5*m+6)))shr (m*5+1)
240      next
250    next
260    b=(g_dat(0,blue)+g_dat(1,blue)+g_dat(2,blue)+g_dat(3,blue))¥4
270    r=(g_dat(0,red)+g_dat(1,red)+g_dat(2,red)+g_dat(3,red))¥4
280    g=(g_dat(0,green)+g_dat(1,green)+g_dat(2,green)+g_dat(3,green))¥4
290    pset(x,y,rgb(r,g,b))
300  next
310 next
```


面を1/4ずつに分割して2月号の拡大ルーチンにかけ（自然画でなければ閾値 t を多少大きくしたほうがいい）、今月の縮小ルーチンで縮めて4つ並べるだけですむ。

労を惜しまず最高のものを得たいなら、適当に下描きした絵を4分割して拡大修正し、また縮小するという手もある。これならふつうのグラフィックツールを使って処理できる。

2Dグラフィックの今後

もともとは取り込み画像に色をつけようということから始まった。

まずは高橋哲史君が編集室のスキヤナを使って取り込んだ元絵に色をつけようと苦戦している図を想像してもらいたい。ペイントしようとしても途中で止まってしまう。今度はモノクロ2階調で取り込んでペイントしてみる。ちゃんとペイントできるが悲しいくらい絵が粗い。

その場は、2値化して取り込んだ輪郭線を細くして色を塗り、その上に多値化された綺麗な輪郭線を合成する、という方法で落ち着いた。そして、考えられたのが丹氏が多階調境界対応のペイントルーチンだった。

その後、福原君の手作業によるアンチエイリアシングを見るにつけ、通常のグラフィックツールの限界と可能性を思い知った。確かに境界線を綺麗に処理してやると非常に高画質な絵が得られることはわかった。しかしそれを手作業で行うというのはあまり

に非人間的な作業だろう。これはある程度自動化できそうな処理に思われた。

グラフィックツールはいまだにZ'sSTAFFを最高峰にしたまま進化が止まっている。Z'sSTAFFがよくできたグラフィックツールであることは間違いない。しかし、そろそろもっと凄いものが出てきてもいいんじゃないだろうか。

* * *

さて、なにはともあれ、必要になるのは十分なメモリだ。たとえばSX-WINDOWでまともなアプリケーションが出てきたとすると、あっというまにメモリが足らなくなるだろう。Macintoshと違いSX-WINDOWは複数のアプリケーションを同時実行することを基本に作られているのでメモリはいくらあっても余ることはない。

考えてみれば、多くの初代X68000やACEユーザーは4万円近く払って1Mバイトの増設を行ったわけだ。それが最近では2MバイトのRAMボードが4万円台で買えるようになってきている。X68000を2,3年も使い込んだユーザーなら、そろそろ増設を考えてもよい頃だろう。効果を考えれば決して高い買い物ではない。

特にグラフィック関係はメモリを大量に必要とする場合が多い。メモリさえあれば内部バッファを1600万色分取って表示部だけ65536色にするなどの方法でより高画質なものを作れる。現状の65536色というのは使っていて極端に不足を感じさせる色数ではない。

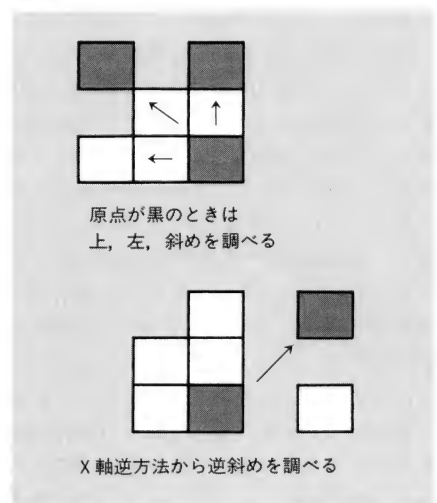
グラフィックツールでグラデーションを

かけたときやレイトレーシングなどを行ったとき以外不足に感じることはないと思う。どちらも表示関係のルーチンをなんとかすれば、65536色でもかなり自然な表示ができるはずだ。うまくやればグラデの縞模様もマッハバンドも出ない。それは今回鈴木氏の256色化や6月号のSXCONVの16色化を見てもわかると思う。

同様に256色モードでも内部で多色処理すればもっと高度なグラフィックツールができるはずだ。しかし、問題は65536色で描いて変換したほうが綺麗だということだろうか。

65536色モードのデータなら扱いやすくほかのモードへの変換も容易(?)だろう。今後の標準はやはりPIC形式の65536(32768)色となるのだろうか？

図4



リスト3

```
10 screen 1,3,1,1
20 str na
30 int i,j,k(3),l,m,n,o,pl,col(3,2),b,r,g,t=8
40 input na
50 pic_load(na+".pic",0,0)
60 for i=0 to 255
70   for j=0 to 255
80     k(0)=point(255-j,255-i)
90     k(1)=point(255-j,255-i-1)
100    k(2)=point(255-j-1,255-i)
110    k(3)=point(255-j-1,255-i-1)
111    pset(511-j*2,511-i*2,k(0))
120    if k(0)=0 then {
130      if k(1)=0 then pset(511-j*2,511-i*2-1,0) else pset(511-j*2,511-i*2-1,k(0))
132      if k(2)=0 then pset(511-j*2-1,511-i*2,0) else pset(511-j*2-1,511-i*2,k(0))
134      if k(3)=0 then pset(511-j*2-1,511-i*2-1,0) else pset(511-j*2-1,511-i*2-1,k(0))
135    } else {
140      pset(511-j*2,511-i*2-1,k(0))
142      pset(511-j*2-1,511-i*2,k(0))
144      pset(511-j*2-1,511-i*2-1,k(0))
149    }
150   next
160   for j=0 to 255
180     k(0)=point(255-j,255-i)
190     /* k(1)=point(255-j,255-i-1)
200     /* k(2)=point(255-j+1,255-i)
210     k(3)=point(255-j+1,255-i-1)
220     if k(0)=0 and k(3)=0 then pset(511-j*2+1,511-i*2-1,0)
230   next
240 next
250 pic_save("ex_"+na,0,0,511,511)
260 input i
270 end
```


色数の補間と量子化

グラフィックデータを変換する

Suzuki Yasuhiro **鈴木 康弘**

X 68000にはいくつかの種類の画面モードが存在します。そのなかでも、グラフィックにもっとも適しているのは、やはり512×512ドットの65536色モードでしょう。Z's STAFF PRO-68Kが扱うのも、この画面モードですし、PICなどの圧縮ツールもこの画面モード専用です(最近、ほかの画面にも対応しているAPICというものもあるが)。

ところで、SX-WINDOWが対応しているグラフィックの画面モードは、768×512ドットの16色モードです。ちまたにあふれている、PC-9801などのグラフィックデータは、640×400ドットの16色で描かれています。これらの16色のグラフィックデータと、65536色のデータで決定的に違うことは、16色のデータは、タイリングを用いて中間色を表現しているのに対し、65536色のデータは、タイリングを用いず、中間色はそのままドットの色となっていることです。

今回のプログラムは、これらの16色で描かれたグラフィックデータを、512×512ドットの65536色モードのデータに変換を試みたものです。ただし、そのまま変換すると、タイリングされたまま65536色のデータになってしまう(当然65536色中の16色しか使わない)、全然65536色を使っている気分になりません。

また縦横比を調節すると(640×400を512×512に変換するので、1ドットの大きさが変わってくる)、タイリングパターンが崩れてしまい、元のデータよりも汚くなってしまう。そこで、タイリングで塗ら

れた領域を、なんとかしてそれに対応する色に変換しなければなりません。

逆に色数の多い画面モード用のデータは色数の少ないモード用にコンバートするアルゴリズムは広く知られていますので、それらを使って65536色のデータを256色モードのデータに変換するプログラムも作ってみました。256色モードはグラフィック画面が2枚あり、どうしてもグラフィック画面が1枚では足りないような場合に威力を発揮します。

こっちのほうは、以前Oh!Xで紹介された、オーダーディザ法と、桑野雅彦氏がプリンタのハードコピー用に考え出されたアルゴリズムを応用したものをういています。また、使われている色数が256色以下の場合、わざわざディザ法を用いるまでもなく256色モードに変換できるので、その処理を行うこともできます。そのほか、画面中でもっともよく使われている256色を抜き出し、それ以外の色をもっとも近い256色で置き換えるというアルゴリズムも発表されていましたが、今回はそれには対応していません(Oh!X1988年2月号参照)。

ちなみに、65536色に変換するとか書いてありますが、実は32768色に変換します(輝度ビットを無視しています)。また、65536色のデータを256色に変換するのではなく、32768色のデータを256色に変換します(輝度ビットのみ異なる色は、同じ色とみなしています)。

コンパイルの方法

プログラムはC言語で書かれています。したがって、XCが必要になるわけですが、XCでコンパイルされたものはとんでもなく処理速度が遅いのです。そこで、Oh!Xの6月号の特別付録にGCCが掲載されているので、できるだけこっちのほうでコンパイルしてください。

GCCでのコンパイル方法は、
gcc T2F.c -O -fstrength-reduce -fo

グラフィックモードの違いを埋める処理に挑戦してみましょう。PC-9801などに描かれた16色のグラフィックデータの色数を増やしてX68000の65536色のデータに変換したり、65536色のデータをできるだけ原画に忠実な256色に変換する際に必要な処理を考えてみます。

```
mit-frame-pointer -liocs -ldos
gcc to256.c -O -fstrength-reduce -fomit-frame-pointer -liocs -ldos
です。ちなみに、XCのほうは、
cc T2F.c -O -Y
cc to256.c -O -Y
です。
```

使い方

まず、16色を32768色に変換するT2F.xですが、あらかじめ、画面を16色モードに設定し(実画面のサイズは1024×1024にしてください)、グラフィックデータを表示しておいてください。T2F.xは、VRAMにあるデータを変換します。そして、グラフィックデータが640×400ならば、

```
T2F -S640
データが512×512ならば、
T2F -S512
```

としてください。これでとりあえず変換を開始します。

また、タイリングパターンを認識して中間色に変換していくので、認識するタイリングパターンの最大値を指定することができます(省略すると、2ドットになります)。たとえば、640×400ドットのデータで、タイリングパターンの最大値を4ドットにするならば、

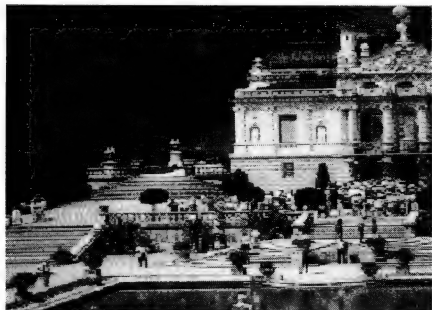
```
T2F -S640 -T4
となります。
```

この、タイリングパターンの最大値というのは、最大何ドットでタイリングされているか、というものです。たとえば、

黒白白黒白白……

というタイリングがある場合には、最大値に3以上を設定しなければ、これはタイリングとみなされず、そのまま残ってしまいます。

この値をむやみに大きくすると、タイリングでないところまでタイリングとみなしてしまい、変なところが1色で塗られてしまいますから、注意してください。



オーダーディザ法による変換

次に、32768色のデータを256色に変換する、to256.xですが、これもグラフィックを表示させてからプログラムを実行させていただきます。

使い方は、スイッチに、オーダーディザ法で変換する場合には“-D”，乗野式アルゴリズムで変換する場合には“-K”，色数を数えて、256色以下の絵をそのまま変換する場合には“-C”をつけ加えて起動してください。

オーダーディザ法で変換する場合には、^{しきい}閾値を指定することができます。たとえば、閾値に60を設定したいのなら、

to256-D60

のように、“-D”に続けて閾値を書きます。省略すると40が設定されます。この値はグラフィックの内容によって最適な値が変わるので、いろいろ試してください。

タイリングについて

16色モードのグラフィックは、ほとんどがタイリングという手法を用いています。このタイリングというのは、たとえば、赤と青のドットを交互に並べていくと、遠目には紫色に見えてしまう、というものです。これを用いると、16色しか出ないはずなのに、それ以上の色を表現することができるのです。

●16色→32768色

まず、タイリングされているグラフィックデータをよ～く見てみますと、タイリングが施されている部分はかなりの規則性があることがわかります。つまり、ある決まったドットの並びが横にず～っと並んでいるのです。色が変わる部分というのは、その決まったドットの並びに合わなくなる部分なのです。

さて、この変換の大まかなアルゴリズムを説明します。

まず、最初にタイリングパターンを横方向に比較していき、そのタイリングパターンが崩れたドットに、フラグを立てて覚えておきます。この処理を全画面に行うと、タイリングパターンが変化した部分（要するに、遠くから見たときの、色が変わる部分→輪郭）にフラグが立つことになります。

あとは、このフラグとフラグのあいだを、その中のタイリングパターンの色で塗ってあげればよいのです。

この変換の核となるタイリングパターンが変化した部分の認識ですが、次の手順で行っています。

- 1) あるドットから右にnドット分を配列変数に格納する
- 2) さらに、その右nドットが、配列変数に格納した色と同じかどうかを調べる
- 3) 同じならば、そこからタイリングパターンが続いていることになる
- 4) 違うのならば、nにn-1を設定して、もう1回調べなおす（1に飛ぶ）
- 5) nが1になってしまったら、そのドットからはタイリングは始まっていない。したがって、そこにフラグを立てて、1ドット右に移動し、新たに調べ始める（1に飛ぶ）

これで、タイリングパターンが続いているかどうかわかります。これがわかったら、次はどこまで続いているかです。これは、次々に配列の内容と実際のドットとを比べていき、それらが異なるところまでとなります。

例を出してみると、

座標 0 1 2 3 4 5 6 7 8 9

色 赤 青 赤 青 赤 青 赤 青 黒 黄

10 11 12 13 14

赤 黄 赤 黄 赤 ……

というドットの並びの場合、まず、先頭の赤青を配列変数に入れます（タイリングの最大値が2の場合）。次に、座標2からの2ドットが、配列変数に入っているものと同じかどうかを調べます。この場合は同じですので、これで赤青というタイリングがあると認識します。

次に、2ドット右に進んで、配列の内容と同じかどうかを調べます。同じですので、さらに2ドット進んで調べます。

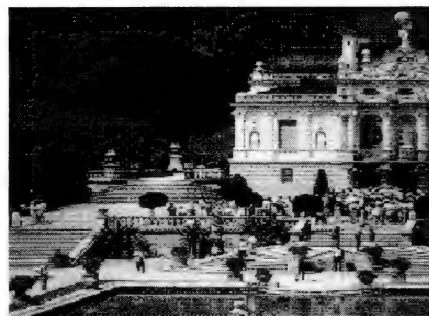
どんどん右に調べていくと、座標8の部分で、配列変数の内容と食い違う色が出てきます。そこで、この座標8のドットにフラグを立てます。

さらに、そこから2ドットを配列に入れます。この場合、黒 黄 が入ります。ところがいきなり次の2ドットと色が異なるため、この黒から始まるのはタイリングではないとみなし、座標9の黄色にフラグを立てます。

そして、次の2ドット（黄 赤）を配列に入れ、再び調べ始めます。

この場合、最初に2ドットを配列に入れて調べましたが、この数値は変更することができ、たとえば3ドットにしてあると、まず3ドットを配列に入れ、そのパターンが続かなければ2ドットにして調べるようになっていきます。

あと、タイリングパターンから色を決める方法ですが、これは単純に、各タイリ



乗野式アルゴリズムによる変換

グパターンのドットのRGB成分の平均を出し、そのRGBの平均によって作られる色になります。

2ドットのタイリングでしたら、

求める色 = (色1 + 色2) ÷ 2

になります。

●640×400→512×512

実は、今回のプログラムでは、あまりにも色の決定の部分のアルゴリズムの部分に時間をさいてしまい、縦横比の変換はかなりいい加減になっています。したがって、640×400のグラフィックを変換しても、そんなに綺麗にはなりません。

具体的にどうやっているかをばらしますと、まず640×400の32768色が記憶できるバッファを取り、タイリングパターンを調べて色を塗るところまでは、そのバッファに対して処理を行います。その後、画面に512×512で表示する段階になったら、1ドットずつ、対応するドットを調べて、それを画面に表示しているのです。したがって、横方向はところどころドットが抜けて表示され、縦方向はところどころ同じドットが2ドット続きます。

この、抜けたり、2ドット続いたりするのが輪郭の部分だったりすると、輪郭が抜けたり、太くなったりしてしまいます。試しに、640×400のグラフィックでも、512×512で変換してみてください。とりあえず、輪郭は綺麗に変換されると思います。

縦横比を調節すると、輪郭が太くなってしまうというのも欠点ですが、まだあります。

このプログラムでは、横方向しかタイリングを調べていないので、たとえば、

赤 白

白 白

という、2×2のタイリングパターンには無力です。

実は、このプログラムの最初のバージョンでは、縦方向も調べていたのですが、横方向で調べた輪郭を認識しない部分が出てくるなど、いろいろ問題点が多かったので

す。そこで、これなら縦方向を無視したほうがいいだろうと思い、横方向のみとなつたわけです。

●32768色→256色

こちらのプログラムでは、オーダードディザ法と桑野氏のアルゴリズムのどちらかで変換できます。それぞれの詳しい原理などは、以前のOh!Xに載っています。オーダードディザ法は、1988年9月号で丹氏が、桑野式アルゴリズムは、1988年11月号で桑野氏が説明しています(1990年6月号にも掲載されている)から、そちらも参照してください。

オーダードディザ法については電腦倶楽部に最近掲載されたものとアルゴリズムから参考文献まで同じですので、同様の実行結果になるようです。

桑野式(もしかしたら、これが誤差拡散法なのだろうか?)は画面の情報量を減らさずに色変換をする優れたアルゴリズムです。たいていの場合、ディザ法を使うよりも自然

な仕上がりになるようです。

RGBという分け方で見ると、256色というのは半端な値なのですが、ここでは6月号のSXCONV(これは65536色を16色に変換する)と同様な考え方に基き、

GGGRRRBB

と、もっとも輝度の低そうな青成分を2ビット、ほかを3ビットで処理することによって自然な色に変換しています。ちなみにX68000標準のパレット設定では、

GRRRRBBB

のように、緑が2ビットで処理されています。

色数が256色以下のグラフィックについては、パレットを変更することによりそのままの画像で256色モードに変換できます。256色しか使われていないグラフィックは少ないように思えるかもしれませんが、レイトレイ取り込みなどを除く、人が描いたようなグラフィックでしたら、たいていの場合256色以下しか使われていません。

おわりに

最初は画面全体にボカシをかけて、色が急に变化する部分を見つけ出し、そこを輪郭として色を塗っていく、という路線で作っていましたが、どうもうまく輪郭が認識できませんでした。

256色に変換するというのも、使われている色数が256色以上の場合は、似た色を同じパレットに割り当てる、という路線で攻めていましたが、いまいち実行速度が遅くなります。なんとかして高速化を図ろうとしたのですが、いつのまにかオーダードディザ法と桑野式アルゴリズムに落ち着いてしまいました。

今回はええいくそ、という掛け声とともに削除されたファイルが数知れず(その直後に、しまった、という掛け声とともに復活されたファイルも数知れず)、なのでした。

リスト1

```
===== T2F.c =====
1: /*
2: /*      16色→32768色 コンバータ version 1.20
3: /*
4: /*      by Yasuhiro Suzuki
5: /*
6: /*
7:
8: #include <stdio.h>
9: #include <stdlib.h>
10: #include <doslib.h>
11: #include <ioclib.h>
12:
13: #define ushort unsigned short
14: #define uint unsigned int
15:
16:
17: /* =====
18: /* グローバル変数の宣言 */
19: /* =====
20: ushort *buf;
21: int xsize = 640;
22: int ysize = 400;
23: int tmax = 2;
24: int pg[16];
25: int pr[16];
26: int pb[16];
27:
28:
29: /* =====
30: /* バッファを初期化する */
31: /* =====
32: void bclr()
33: {
34:     ushort *p = buf;
35:     int i;
36:     for( i=xsize*ysize; i>0; i-- ){
37:         /* 1で初期化しているの
38:         *(p++) = 1;
39:         /* 黒(0)と区別するた
40:     }
41: }
42: /* =====
43: /* パレットをRGBに分解して保存 */
44: /* =====
45: void ptrns()
46: {
47:     ushort *p = (ushort *)0xE82000;
48:     int i, c;
49:     for( i=0; i<16; i++){
50:         c = *(p++);
51:         pr[i] = (c >> 11) & 0x1F;
52:         pg[i] = (c >> 6) & 0x1F;
53:         pb[i] = (c >> 1) & 0x1F;
54:     }
55: }
56:
57:
58: /* =====
59: /* コマンドオプションを調べる */
60: /* =====
61: int chkw( ac, av )
62: int ac;
63: char *av[];
64: {
65:     int i, c;
66:     for( i=1; i<ac; i++){
67:         if( ( av[i][0] == '-' ) || ( av[i][0] == '/' ) ){
```

```
69:         c = av[i][1] | 0x20;
70:         if( c == 's' ){
71:             xsize = atoi( &av[i][2] );
72:             if( xsize == 512 ){
73:                 ysize = 512;
74:             }
75:             else if( xsize == 640 ){
76:                 ysize = 400;
77:             }
78:             else{
79:                 return( 1 );
80:             }
81:         }
82:         else if( c == 't' ){
83:             tmax = atoi( &av[i][2] );
84:             if( tmax == 0 ){
85:                 return( 1 );
86:             }
87:         }
88:         else{
89:             return( 1 );
90:         }
91:     }
92:     else{
93:         return( 1 );
94:     }
95: }
96:
97: return( 0 );
98: }
99:
100: /* =====
101: /* タイリングのドット数を調べる */
102: /* =====
103: int tlen( vp, max )
104: ushort *vp;
105: int max;
106: {
107:     ushort *p, ti[256];
108:     int i, j, r;
109:
110:     if( max > tmax )
111:         max = tmax;
112:
113:     for( i=max; i>1; i-- ){
114:         p = vp;
115:         for( j=0; j<i; j++){
116:             ti[j] = *(p++);
117:         }
118:         r = 1;
119:         for( j=0; j<i; j++){
120:             if( ti[j] != *(p++) ){
121:                 r = 0;
122:                 break;
123:             }
124:         }
125:         if( r ){
126:             return( i );
127:         }
128:     }
129:
130:     return( 0 );
131: }
132:
133: /* =====
134: /* 境界色を書き込む */
135: /* =====
136: void tset( x, y, ti, n )
137: int x, y, n;
138: ushort *ti;
139: {
140:     int i;
```



```

141:   uint   g, r, b, c;
142:
143:   g = r = b = 0;
144:   for( i=0; i<n; i++, ti++ ){
145:       g += pg[ti];
146:       r += pr[ti];
147:       b += pb[ti];
148:   }
149:
150:   g /= n;
151:   r /= n;
152:   b /= n;
153:   c = ( g << 11 ) | ( r << 6 ) | ( b << 1 );
154:
155:   *( buf + (int)( x + y * xsize ) ) = c;
156: }
157:
158: /*****
159:  * タイリングを調べる */
160: ******/
161: void tilex()
162: {
163:     ushort ti[256];
164:     ushort *vp, *vvp;
165:     int     x, y, t, i;
166:
167:     vvp = (ushort *)0xC00000;
168:     for( y=0; y<ysize; y++ ){
169:         t = 0;
170:         vp = vvp;
171:         vvp += 1024;
172:         for( x=0; x<xsize; ){
173:             if( t == 0 ){
174:                 if( t = tlen( vp, ( xsize - x ) / 2 ) ){
175:                     for( i=0; i<t; i++ ){
176:                         ti[i] = *(vp++);
177:                     }
178:                     tset( x, y, ti, t );
179:                     x += t;
180:                 }
181:                 else{
182:                     ti[0] = *(vp++);
183:                     tset( x++, y, ti, 1 );
184:                 }
185:             }
186:             else{
187:                 if( t > ( xsize - x ) ){
188:                     t = xsize - x;
189:                 }
190:                 for( i=0; i<t; i++, x++, vp++ ){
191:                     if( ti[i] != *vp ){
192:                         t = 0;
193:                         break;
194:                     }
195:                 }
196:             }
197:         }
198:     }
199: }
200:
201: /*****
202:  * 境界色で塗り潰す */
203: ******/
204: void fullx()

```

```

205: {
206:     ushort *bp, *vp, c;
207:     int     x, y, xx, yy;
208:
209:     bp = buf;
210:     for( y=0; y<ysize; y++ ){
211:         c = *bp;
212:         for( x=0; x<xsize; x++ ){
213:             if( *bp != 1 ){
214:                 c = *(bp++);
215:             }
216:             else{
217:                 *(bp++) = c;
218:             }
219:         }
220:     }
221:
222:     vp = (ushort *)0xC00000;
223:     for( y=0; y<512; y++ ){
224:         yy = ( ( y * ysize ) / 512 ) * xsize;
225:         for( x=0; x<512; x++ ){
226:             xx = ( x * xsize ) / 512;
227:             *(vp++) = *( buf + (int)( xx + yy ) );
228:         }
229:     }
230: }
231:
232: /*****
233:  * メインルーチン */
234: ******/
235: int main( ac, av )
236: int     ac;
237: char    *av[];
238: {
239:     puts( "TILE to FULL ver1.20 by Yasuhiro Suzuki" );
240:
241:     if( chksw( ac, av ) ){
242:         puts( "[使用法] T2F [ <スイッチ> ] ..." );
243:         puts( "Tt-S540: 640 x 400 ドットの絵を変換する。" );
244:         puts( "Tt-S512: 512 x 512 ドットの絵を変換する。" );
245:         puts( "Tt-Tn: 識別するタイリングパターンのドット数の最大値" );
246:         return( 1 );
247:     }
248:
249:     if( ( buf = (ushort *)MALLOC( xsize * ysize * 2 ) ) >= (ushort *)0
x80000000 ){
250:         puts( "メモリが足りません。" );
251:         return( 1 );
252:     }
253:
254:     SUPER(0);           /* スーパーバイザモードになる */
255:
256:     bclr();             /* バッファを初期化する */
257:     ptrns();            /* パレットを保存 */
258:
259:     tilex();            /* タイリングの変化点を抽出する */
260:
261:     CRTMOD( 12 );       /* 画面を初期化する */
262:     G_CLR_ON();
263:
264:     fullx();            /* V R A M に表示する */
265:
266:     return( 0 );
267: }

```

リスト2

```

===== to256.c =====
1: /*
2:  *      6 5 5 3 6 色 → 2 5 6 色 コンバータ version 2.13
3:  *
4:  *      by Yasuhiro Suzuki
5:  *
6:  */
7:
8: #include <stdio.h>
9: #include <stdlib.h>
10: #include <locale.h>
11: #include <doslib.h>
12:
13: #define uchar   unsigned char
14: #define ushort  unsigned short
15:
16: #define VRAM     (ushort *)0xC00000
17:
18:
19: /*****
20:  * グローバル変数の宣言 */
21: ******/
22: uchar    *buf;
23: ushort    pcnv[32768];
24: int       dn;
25: int       br[2][512];
26: int       bb[2][512];
27: int       mat[8][8] = {
28:     0, 32, 8, 40, 2, 34, 10, 42,
29:     48, 16, 56, 24, 50, 18, 58, 26,
30:     12, 44, 4, 36, 14, 46, 6, 38,
31:     68, 28, 52, 20, 62, 30, 54, 22,
32:     3, 35, 11, 43, 1, 33, 9, 41,
33:     51, 19, 59, 27, 49, 17, 57, 25,
34:     15, 47, 7, 39, 13, 45, 5, 37,
35:     63, 31, 55, 23, 61, 29, 53, 21
36: };
37:
38:
39: /*****
40:  * パレット */
41: ******/
42: #define IM (256*31)
43:
44: int       rrr[7] = { IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
45: int       sss[7] = { IM/7, IM*2/7, IM*3/7, IM*4/7, IM*5/7, IM*6/7,
IM*7/7 };
46: int       bbb[3] = { IM/3, IM*2/3, IM*3/3 };
47:
48: unsigned short  rgb[8][8][4];
49:
50: #define RMAX    31

```

```

51: #define GMAX     31
52: #define BMAX     31
53:
54: #define PALRGB(R,G,B) ( (((R)*GMAX/7)&31)<<11 | (((R)*RMAX/7)
&31)<<6 | (((B)*BMAX/3)&31)<<1 )
55:
56: /*****
57:  * 色数を数える */
58: ******/
59: int count()
60: {
61:     ushort *vp, c;
62:     int     i, k;
63:
64:     for( i=0; i<32768; i++ ){
65:         pcnv[i] = 0;
66:     }
67:
68:     vp = VRAM;
69:     k = 1;
70:     for( i=512*512; i>0; i-- ){
71:         c = *(vp++) >> 1;
72:         if( pcnv[c] == 0 ){
73:             pcnv[c] = k++;
74:         }
75:     }
76:
77:     return( k - 1 );
78: }
79:
80: /*****
81:  * V R A M の内容を256色に変換して格納 */
82: ******/
83: void trns()
84: {
85:     uchar *bp;
86:     ushort *vp;
87:     int     i;
88:
89:     bp = buf;
90:     vp = VRAM;
91:     for( i=512*512; i>0; i-- ){
92:         *(bp++) = pcnv[ *(vp++) >> 1 ] - 1;
93:     }
94: }
95:
96: /*****
97:  * オーダードリザで色を変換する */
98: ******/
99: void dither()
100: {
101:     uchar *bp;
102:     ushort *vp, c;

```



```

103: int g, r, b, d;
104: int x, y;
105:
106: bp = buf;
107: vp = VRAM;
108: for( y=0; y<512; y++ ){
109:     for( x=0; x<512; x++ ){
110:         c = *(vp++);
111:         g = ((c >> 11) & 0x1F) << 3;
112:         r = ((c >> 6) & 0x1F) << 3;
113:         b = ((c >> 1) & 0x1F) << 3;
114:         d = mat[ y & 0x07 ][ x & 0x07 ];
115:         g = ( g + d ) / dn / 2;
116:         r = ( r + d ) / dn;
117:         b = ( b + d ) / dn;
118:         if( g > 3 ){
119:             g = 3;
120:         }
121:         if( r > 7 ){
122:             r = 7;
123:         }
124:         if( b > 7 ){
125:             b = 7;
126:         }
127:         *(bp++) = ( g << 6 ) | ( r << 3 ) | b;
128:     }
129: }
130: }
131:
132: /******
133: /* 桑野式アルゴリズムで変換 */
134: /******
135: void kuwano()
136: {
137:     uchar *bp;
138:     ushort *vp, c;
139:     int x, y, lc, lb;
140:     unsigned int cg, cr, cb, dg, dr, db;
141:     int i;
142:
143:     bp = buf;
144:     vp = VRAM;
145:     for( x=0; x<512; x++ ){
146:         bg[0][x] = br[0][x] = bb[0][x] = 0;
147:     }
148:     for( y=0; y<512; y++ ){
149:         lc = y & 1;
150:         lb = ( y + 1 ) & 1;
151:         for( x=0; x<512; x++ ){
152:             bg[lb][x] = br[lb][x] = bb[lb][x] = 0;
153:         }
154:         cg = cr = cb = 0;
155:         for( x=0; x<512; x++ ){
156:             c = *vp;
157:             cg += ((c >> 11) & 0x1F) * 256 + bg[lc][x];
158:             cr += ((c >> 6) & 0x1F) * 256 + br[lc][x];
159:             cb += ((c >> 1) & 0x1F) * 256 + bb[lc][x];
160:
161:             dg=dr=db=0;
162:             for( i=6; i>=0; i-- ){
163:                 if( cg >= ggg[i] ){
164:                     dg = i+1;
165:                     cg -= ggg[i];
166:                     break;
167:                 }
168:             }
169:             for( i=6; i>=0; i-- ){
170:                 if( cr >= rrr[i] ){
171:                     dr = i+1;
172:                     cr -= rrr[i];
173:                     break;
174:                 }
175:             }
176:             for( i=2; i>=0; i-- ){
177:                 if( cb >= bbb[i] ){
178:                     db = i+1;
179:                     cb -= bbb[i];
180:                     break;
181:                 }
182:             }
183:
184:             *(bp++) = ( dg << 5 ) | ( dr << 2 ) | db;
185:             *(vp++) = PALRGB( dr, dg, db );
186:
187:             bg[lb][x] += cg/8;
188:             br[lb][x] += cr/8;
189:             bb[lb][x] += cb/8;
190:             bg[lb][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cg/4);
191:             br[lb][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cr/4);
192:             bb[lb][ ( x > 0 ) ? ( x - 1 ) : ( x ) ] += (cb/4);
193:             bg[lb][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += (cg/8);
194:             br[lb][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += (cr/8);
195:             bb[lb][ ( x < 511 ) ? ( x + 1 ) : ( x ) ] += (cb/8);
196:
197:             cg/=2;
198:             cr/=2;
199:             cb/=2;
200:         }
201:     }
202: }
203: /******
204: /* パレットを設定 */
205: /******
206: void setpal()
207: {
208:     ushort *pp, c;
209:     int i;
210:
211:     pp = (ushort *)0xE82000;
212:     for( i=0; i<32768; i++ ){
213:         if( (c = pconv[i]) != 0 ){
214:             *(pp + c - 1) = i << 1;
215:         }
216:     }
217: }
218:
219: /******
220: /* バッファの内容をVRAMに転送 */
221: /******
222: void prt()
223: {
224:     uchar *bp;
225:     ushort *vp;

```

```

226: int i;
227:
228: bp = buf;
229: vp = VRAM;
230: for( i=512*512; i>0; i-- ){
231:     *(vp++) = (ushort)*(bp++);
232: }
233: }
234:
235: /******
236: /* 画面モード初期化 */
237: /******
238: void ginit0()
239: {
240:     CRTMOD( 8 );
241:     G_CLR_ON();
242: }
243:
244: void ginit1()
245: {
246:     int r, g, b;
247:
248:     CRTMOD( 8 );
249:     G_CLR_ON();
250:
251:     for( g=0; g<8; g++ ){
252:         for( r=0; r<8; r++ ){
253:             for( b=0; b<4; b++ ){
254:                 rgb[r][g][b] = (g<<13|r<<8|b<<4);
255:                 GPALET( (g<<5|r<<2|b), PALRGB( r, g, b ) );
256:             }
257:         }
258:     }
259:     return;
260: }
261:
262: /******
263: /* コマンドオプションをチェック */
264: /******
265: int chksw( ac, av )
266: int ac;
267: char *av[];
268: {
269:     int c, i, r=0;
270:
271:     if( ac < 2 ){
272:         return( 0 );
273:     }
274:
275:     for( i=1; i<ac; i++ ){
276:         if( av[i][0] != '-' ){
277:             return( 0 );
278:         }
279:         c = av[i][1] | 0x20;
280:         switch( c ){
281:             case 'd': r |= 0x01;
282:                     dn = atoi( &av[i][2] );
283:                     if( dn == 0 ){
284:                         dn = 40;
285:                     }
286:             case 'k': r |= 0x02;
287:                     break;
288:             case 'c': r |= 0x04;
289:                     break;
290:             default: return( 0 );
291:         }
292:     }
293:
294:     if( (( r & 0x03 ) == 0x03 ) || (( r & 0x03 ) == 0x00 ) ){
295:         return( 0 );
296:     }
297:
298:     return( r );
299: }
300:
301:
302: /******
303: /* メインルーチン */
304: /******
305: int main( ac, av )
306: int ac;
307: char *av[];
308: {
309:     int m;
310:
311:     puts("65536 to 256 ver.2.13 by Yasuhiro Suzuki");
312:
313:     if( ( buf = (uchar *)MALLOC( 512 * 512 ) ) >= (uchar *)0x00000000 )
314:     {
315:         puts("メモリが足りません。");
316:         return( 1 );
317:     }
318:
319:     if( !( m = chksw( ac, av ) ) ){
320:         puts("[ 使用法 ] to256 <スイッチ>");
321:         puts("Yt-DnYtオーダードディザ法で変換を行う ( n はしきい値 )");
322:
323:         puts("Yt-KYt桑野式変換を行う");
324:         puts("Yt-CYt色数を調べて変換する");
325:         return( 0 );
326:     }
327:
328:     SUPER(0);
329:
330:     if( ( m & 0x04 ) && ( count() <= 256 ) ){
331:         trns();
332:         ginit0();
333:         setpal();
334:     }
335:     else if( m & 0x01 ){
336:         dither();
337:         ginit0();
338:     }
339:     else{
340:         kuwano();
341:         ginit1();
342:     }
343:
344:     prt();
345:
346:     return( 0 );
347: }

```


4096色→8色変換

Zの画像をX1で

Kameda Masahiko 亀田 雅彦

なぜ、8色なの?

今月は大盤振る舞いなのです。まさに「もってけどろぼう!」の世界といえるでしょう。なぜかという、この特集とKAME-DOS連載の豪華2本立てだからです。しかも、それが見事に調和を保ちながらダブル進行していくという華麗さ、名づけて「シンクロ原稿」です。「ライターがX1関係で荒稼ぎをしようとしてる」とか、「1本のプログラムを使い回してるだけだ」という噂の真偽はさておき、特集とは名ばかり、KAME-DOS関係の話が割り込んでくるので悪しからず。

しかしながら、グラフィック特集である以上グラフィックにも力をいれなければなりません。そこで今回は「Zの4096色画像を8色に変換してみよう」ということになりました。ここでふと思い浮かんでくるのは、6月号のSX-WINDOWのグラフィックについて。パラパラとめくってみると、そのものずばり載っているじゃあないですか。しかもその6月号ですら、1988年11月号の引用なのだから、私は「引用の引用」をするという、神をも恐れぬワザにでようというわけです。でも楽しいことはいいことなので、そのまま採用させてもらいました(実際の実行結果も良好でした)。

それじゃ8色に変換してうれしいこと。

●メモリが節約できる

96K(4096色フル)だと2Dディスクで3枚ちょっと。2HD(アクセスが遅い)なら10枚くらいで、結構邪魔です。ディスクアクセス側の問題もありますが、容量はロード時間にも影響を与えます。

●互換性が出てくる

4096色というのはあまりメジャーな数字ではないですが、8、16色あたりはMS-DOSの世界では常識です。もちろんX1のVRAMデータ形式のままでは互換性はありませんが、変換自体は簡単にできそうなので挑戦してみるのも面白そうです。

●プリンタとの相性がいい

実はこれが一番身近な問題だと思います。Zではアナログ画像取り込みが標準で装備されながら、あまり活用されないのはグラフィックの扱いにくさが原因でしょう。カラーイメージボードは8色でありながら、そのデータの少なさがよいほうへ働いています。Z標準のアナログ画像を精一杯有効に活用していきたいとすれば、8色に落としてプリンタへの出力を容易にするのが効果的です。ひょっとすると、安価なスキャナとしての価値をZに見出せるかもしれません。

もちろん、8色にして悪いことは原画の情報がRGBの各色について1/4ずつになることです。これをなるべく緩和しようとするのが前述のアルゴリズムです。

プログラムは?

画像変換のためのものと画面ローダ、画面セーバの3本を用意しました。ローダとセーバに関しては、連載のKAME-DOSの外部コマンドとしても使えるようになっていきます(そっちがメインだったりして)。もともとKAME-DOSのほうで外部コマンドの許容範囲が広いので、画像変換プログラムもコマンドとすることができます(あまり意味はありません)。

画像変換プログラムは人のアルゴリズムを使っているものであまり自慢できたものじゃありませんが、ローダとセーバのスピードに関しては自信を持っています。KAME-DOSの実力をいかに発揮させて、理論的な最高速に達しました。画面全体を一度にロードしたりセーブしたりしかできませんが、そのスピードは一度見てもらえればわかります。

こんなスピードを競うようなプログラムは最近では見かけませんが、8ビット全盛の頃はよくはやったものです(特にグラフィック命令)。自分で書いててなつかしくなりました。

入力方法

画像変換処理のX1シリーズでの応用例です。X1turboZの4096色画像を「楽野式アルゴリズム」で8色のデータに変換してみましょう。同時にKAME-DOS上でグラフィックを扱うためのコマンドについても解説します。今回のINTEGRAL X1の連載記事もあわせてご覧ください。

●画像変換プログラム(リスト1&リスト2)

X1turboZでなおかつZ-BASIC専用プログラムです(必ずしもKAME-DOSは必要ありません)。Z-BASICからリスト1を入力したら、ファイル名はとりあえず「CCHANGE.X1」としてセーブしておいてください。次に、CLEAR &HC000を実行して、リスト2をなんらかのマシン語入力ツールから打ち込んでください。間違いがなければSAVEM「CCHANGE.OBJ」&HC000,&HC1A7としてセーブします。使うときは両方必要になるので、2つは同一ディレクトリ上においてください。

●画面ローダ

●画面セーバ

X1全シリーズで使うことができます。ただし、6月号から今月にかけて連載しているKAME-DOSシステムが必要になります。具体的には、「INTEGRAL.X」「COMMAND.X1」「FDC.OBJ」の3つのプログラムと、ノーマルX1には7月号のプログラムも必要です。まだ持っていない方は、バックナンバーなどからぜひ入手してください。

「CZ-8FB01,turboBASIC,Z-BASIC」のうちKAME-DOSのあるBASICで、今月の92ページから連載に載っているリストを入力します。「COMMAND.X1」と同一ディレクトリ上にセーブしてください。変数名の間違いがあったりすると、ディスクを破壊しかねないので慎重にチェックしてください。

また、入力上の注意は今月号の連載の「外部コマンド」の入力法のところをよく読んで必ず守るようにしてください。ファイル名はそれぞれ「GLOAD.X1」「GSAVE.X1」とします。

●まとめ

1:Z-BASIC&KAME-DOS

「COMMAND.X1」「CCHANGE.X1」
「CCHANGE.OBJ」「GLOAD.X1」
「GSAVE.X1」を同一ディレクトリ上にお
いてください。

2:Z-BASICのみの方

「CCHANGE.X1」「CCHANGE.OBJ」
を同一ディレクトリ上においてください。

3:KAME-DOSのみの方

「COMMAND.X1」「GLOAD.X1」
「GSAVE.X1」を同一ディレクトリ上にお
いてください。

使い方

●CCHANGE.X1 (KAME-DOSなし)

あらかじめグラフィックを表示させてお
いてCCHANGE.X1を起動します。メニュ
ー画面になるので、1を押すと全画面に対
して(少し時間がかかります)4096色か
ら8色へ変換します。それが終わると、キ
ー入力待ちになって、入力するとメニュー
へ戻ります。メニューの2、3は使えませ
ん。4で終了です。

スペースキーでグラフィックのON/

OFFができます。

●CCHANGE.X1 (KAME-DOSあり)

KAME-DOSのコマンドライン ([X:/])
から「CCHANGE」として起動します。
GLOAD.X1,GSAVE.X1があればメニュ
ーの2、3が使えます。それぞれ選択する
と、ファイル名の入力になるので、ドライ
ブ名を含めてフルパスで指定してくださ
い。リターンキーだけを押せば、メニューに
戻ります。その後の操作はGLOAD,GSAVE
と同じになります。

メニューからグラフィックをロードする
こともできますが、あらかじめロードし
ておきたいこともあります。そういうときは、
グラフィックをロードして、「INTEGRAL.
X」の中のグラフィックを消すような命令
を削ってから、KAME-DOSを起動してく
ださい。

●GLOAD.X1

96Kバイトあるいは64Kバイト(自動的
に判断する)のグラフィックファイルをロ
ードします。ファイルの拡張子によって画
面モードを自動変更するので注意してく
ださい(図1)。

KAME-DOSのコマンドラインから
「GLOAD ファイル名」として起動しま
す。エラーがなければ、グラフィックを表
示してキー入力待ちになるので、キーを押
すと親プロセスへもどります。エラーがあ
ればメッセージを表示して実行を中止しま
す。ロードしている最中は少しキャラクタ
画面が乱れますが、それが正常なので心配
いりません。ノーマルX1では96Kファイ
ルはロードできません。

Z-BASICには標準でVLOAD,VSAVE

INTEGRALXを書き換える

次に挙げる命令を、自分のINTEGRAL.Xで削
除してください。ただし、これはグラフィッ
クをあらかじめロードしておいたときのみの
処置なので、通常はいつものINTEGRAL.Xを使
ってください。

1040行のWIDTH・1050行のCLS 4

1200行のINIT

また、INTEGRAL.XやCOMMAND.X1にある
SCREEN命令はグラフィック画面を見えなく
するものなので、必要に応じて入れておいて
ください。Z-BASICを使う場合は、OPTION
SCREEN 4をOPTIONSCREEN 5に換えておき
ましょう。

リスト1

```
1000 'CCHANGE.X1 Ver 1.0 By Kameda
1010 '
1020 OPTIONSCREEN 4:WIDTH 40,25,0,1:OPTIONSCREEN 5:INIT:DEFINT a-z
1030 DEFUSR0=m_tranr
1040 '
1050 CLEAR &HC000:LOADM "CCHANGE.OBJ"
1060 '----- ( MAIN ROUTINE ) -----
1070 '
1080 SCREEN:CLS:fe$(1)="":GOSUB "menu":CLS
1090 ON a GOTO 1100,"load","save",1160:IF a$=CHR$(27) GOTO 1160
1100 '
1110 KLIST 0:CONSOLE 0,25
1120 MEM$(&HC007,8)=MKI$(0)+MKI$(320)+MKI$(0)+MKI$(200)
1130 OPTIONSCREEN 4:INIT:CFLASH 1:PRINT "Wait a moment.":CFLASH 0
1140 CALL &HC000
1150 GOSUB "ending":GOTO 1080
1160 '
1170 CLS:IF proces=0 THEN END
1180 CLEAR &HD000:proces=proces-1:CHAIN proces$(proces)
1190 '----- ( LOAD ) -----
1200 '
1210 LABEL "load"
1220 IF proces=0 THEN 1080
1230 LOCATE 7, 7:PRINT "*** GRAPHIC LOAD ***"
1240 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1250 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME>";:COLOR 7:INPUT "",fe$(1)
1260 IF fe$(1)=" " THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1270 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GLOAD.X1"
1280 '----- ( SAVE ) -----
```

```
1290 '
1300 LABEL "save"
1310 IF proces=0 THEN 1080
1320 LOCATE 7, 7:PRINT "*** GRAPHIC SAVE ***"
1330 LOCATE 11,10:COLOR 1:PRINT "[RETURN]: MENU"
1340 LOCATE 5,13:COLOR 6:PRINT "FILE-NAME>";:COLOR 7:INPUT "",fe$(1)
1350 IF fe$(1)=" " THEN 1080 ELSE POKE v_wfd0,PEEK(&HF8D6)
1360 proces$(proces)="CCHANGE.X1":proces=proces+1:CHAIN "GSAVE.X1"
1370 '----- ( MENU ) -----
1380 '
1390 LABEL "menu"
1400 LOCATE 3, 5:PRINT "*** 4096 -> 8 color change ***"
1410 LOCATE 10, 8:COLOR 3:PRINT "[1]";:COLOR 7:PRINT "COLOR CHANGE"
1420 LOCATE 10,10:COLOR 3:PRINT "[2]";:COLOR 7:PRINT "GRAPHIC LOAD"
1430 LOCATE 10,12:COLOR 3:PRINT "[3]";:COLOR 7:PRINT "GRAPHIC SAVE"
1440 LOCATE 10,14:COLOR 3:PRINT "[4]";:COLOR 7:PRINT "END"
1450 LOCATE 14,16:COLOR 5:PRINT "push [1]-[4]"
1460 LOCATE 7,18:COLOR 7:PRINT "[SPACE]: graphic ON & OFF"
1470 k=0:REPEAT:a$=INKEY$:a=VAL(a$)
1480 IF a$=" " THEN IF k THEN SCREEN:k=0 ELSE INIT:k=1
1490 UNTIL (1<=a AND a<=4) OR a$=CHR$(27)
1500 RETURN
1510 '----- ( END ) -----
1520 '
1530 LABEL "ending"
1540 CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1550 REPEAT:A$=INKEY$:UNTIL A$<>" "
1560 CLS:CONSOLE 0,24:KLIST 1:RETURN
```

リスト2 CCHANGE.OBJ

```
C000 C3 16 C0 00 00 00 00 00 : 99
C008 00 40 01 00 00 C8 00 2C : 35
C010 14 2C 00 00 00 00 21 00 : 61
C018 C8 11 01 C8 01 FF 07 AF : 58
C020 77 ED B0 2A 0B C0 22 05 : 30
C028 C0 D9 DD 21 00 C0 21 00 : 84
C030 C8 D9 3A 05 C0 E6 01 28 : AF
C038 08 D9 E5 DD E5 E1 DD E1 : 27
C040 D9 DD E5 E1 5D 54 13 01 : 41
C048 FF 03 AF 77 ED B0 D9 ED : 8B
C050 5B 07 C0 DD 19 19 D9 CD : D7
C058 6B C0 2A 05 C0 23 22 05 : 64
C060 C0 ED 5B 0D C0 B7 ED 52 : CB
C068 38 BF C9 AF 32 0F C0 32 : A2
C070 10 C0 32 11 C0 2A 07 C0 : C4
C078 22 03 C0 CD 8F C0 2A 03 : 2E
-----
SUM: 6E 21 02 C9 15 0A 0E F0 B594
```

```
C080 C0 23 22 03 C0 ED 5B 09 : 19
C088 C0 B7 ED 52 38 ED C9 CD : 71
C090 80 C1 FD 21 0F C0 3E 40 : AC
```

```
C098 32 12 C0 CD 02 C1 FD 7E : 0F
C0A0 00 82 D9 86 D9 16 00 FE : CF
C0A8 78 38 04 D6 78 16 01 CD : E6
C0B0 C7 C0 CD 41 C1 FD 23 DD : 53
C0B8 23 D9 23 D9 3A 12 C0 C6 : CA
C0C0 40 32 12 C0 20 D5 C9 5F : 61
C0C8 CB 3F CB 3F CB 3F FD 77 : 92
C0D0 00 7B E6 07 DD 86 00 FD : C8
C0D8 86 00 DD 77 00 2A 03 C0 : C7
C0E0 7C B5 28 04 FD 7E 00 87 : 65
C0E8 DD 86 FD DD 77 FD 7E : 2C
C0F0 00 DD 86 03 DD 77 03 FD : BA
C0F8 7E 00 CB 27 CB 27 FD 77 : D6
-----
SUM: FC 04 AF 47 39 73 09 0E 43E1
```

```
C100 00 C9 CD 9E C1 16 00 CD : D8
C108 25 C1 01 D0 1F ED 78 F6 : 31
C110 10 FD 79 CD 25 C1 01 D0 : FA
C118 1F ED 78 E6 EF ED 79 7A : 39
C120 07 07 07 57 C9 E5 CD 32 : 19
C128 C1 01 00 04 09 CD 32 C1 : 8F
```

```
C130 E1 C9 4D 44 ED 78 A3 28 : 6B
C138 04 37 CB 12 C9 B7 CB 12 : 75
C140 C9 CD 9E C1 CD 5D C1 01 : E1
C148 D0 1F ED 78 F6 10 ED 79 : C0
C150 CD 5D C1 01 D0 1F ED 78 : 40
C158 E6 EF ED 79 C9 E5 CD 6A : 20
C160 C1 01 00 04 09 CD 6A C1 : C7
C168 E1 C9 4D 44 CB 1A 38 0A : 62
C170 D5 7B 2F ED 58 A3 ED 79 : CD
C178 D1 C9 ED 78 B3 ED 79 C9 : E1
-----
SUM: 95 B2 80 32 B7 7A CF A3 31E2
```

```
C180 2A 03 C0 ED 5B 05 C0 AF : A9
C188 32 F6 FB 06 1D ED 41 CD : 41
C190 07 59 06 1E ED 41 22 13 : E7
C198 C0 7A 32 15 C0 C9 3A 12 : 56
C1A0 C0 2A 13 C0 84 67 3A 15 : F7
C1A8 C0 5F C9 : E8
-----
SUM: A3 55 CF E6 A9 63 97 B6 5AB4
```


というグラフィック保存用の命令がありますが、GLOAD,GSAAVEのデータ形式はそのフォーマットとまったく同じです。したがって、VSAVEによってセーブされたファイルはGLOADでロードできるし、その逆もまたしかりです（違いは「速さ」だけ）。「ベタ書きフォーマット」であまり賢くないのですが、これが標準なのでしかたありません。

●GSAVE.X1

グラフィック画面のセーブです。ロードと同じように起動しますが、セーブする前に96Kバイトか64Kバイトにするかを聞いてきます。画像のグラフィックモードに合わせて決定してください。セーブ時もロード時と同じように画面が乱れます。なお、64KファイルはGSAVE独自のものなので、Z-BASICのVLOADではロードできません。

これらのプログラムをKAME-DOS上で使うときには、重要な注意点がひとつあります。よく読んでください。それは、DOSのバッファをG-RAMに設定している場合です（バッファに関しては6, 7月号参照のこと）。画像ファイルをセーブしようとしてディスクアクセスすると、バッファがG-RAM上にあるのでグラフィックが破壊されてしまいます。これでは困るので、バッファをはかに移す必要があるのです。

バンクメモリを搭載していればそこにバッファを設定して一件落着なのですが、そうとは限りません。そこでX1に残された最後の領域であるキャラクタ&アトリビュートエリアに、バッファを設定します（そうです！ このおかげでロード/セーブ時に画面が乱れるのです）。これは一時緊急避難の処置なので、これが終わったらすぐに元へ戻してください。具体的な作業は囲みに書いておきます。

必殺！ アルゴリズム

実は、CCHANGE.OBJ（リスト2）は単独でも使用可能なのです。4096色グラフィックを表示させておいて、CLEAR & HC000:LOADM "CCHANGE.OBJ"でマシン語をロードします。その後、CALL & HC000を実行すれば8色に変換してくれます。これを利用すると、ZでないturboZのアナログ画像データをロードし（GLOAD）、8色に変換して、アナロググラフィックをそれなりに見ることもできます（データが手に入れば、だけど）。

また、CCHANGE.X1内で、MEM\$(

HC007,8)=MKI\$(0)+MKI\$(320)+MKI\$(0)+MKI\$(200)という行があります。このMKI\$の中身は順に左上X座標、右下X座標+1、左上Y座標、右下Y座標+1になっていて、この矩形領域が変換対象になります。書き換えて実行してみるとよくわかると思います。

さて、CCHANGEルーチンのアルゴリズムはバックナンバーを見てもらうとして、ここではGLOADとGSAVEについて解説します。

それぞれKAME-DOSのディスクアクセスルーチンを使っているわけですが、KAME-DOSには標準のG-RAMロード&セーブルーチンはありません。どうしてるのかというと、G-RAM全体(48Kバイト×2)をバッファとみなして、通常はデータの仲介役のバッファに、最初からデータを入れておいたことにします。

もともとバッファの大きさは4Kバイト単位の変長なので48Kバイトでも問題はあります。

この方式の長所としては「BASICでも簡単に制御できる・速くなる」などがあって、短所は「ベタ書きフォーマットにしか通用しない・任意の矩形領域は取り扱いできない」などです。そのためZ'STAFFのフォー

図1 拡張子と画面モードの関係

[X:] GLOAD A:GAZO.GLO

```

GLO: WIDTH 40,25,0,1 4096色モード
GL1: WIDTH 80,25,0,1 64色モード
GM0: WIDTH 40,25,0,2 64色モード
GM1: WIDTH 80,25,0,2 8色モード
GH0: WIDTH 40,25,1,2 64色モード
GH1: WIDTH 80,25,1,2 8色モード
GL2: WIDTH 40,25,0,1 64色2画面

```

(マニュアルより抜粋)

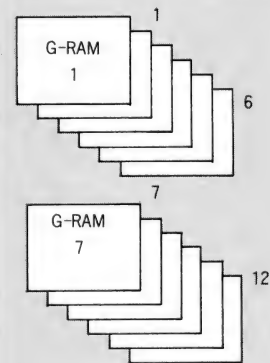
これに新たに、GL3: WIDTH 40,25,0,1 8色モードをつけました。なおGL3の場合は32Kバイトしか使っていませんが、64K分のファイルになります。その他はすべて96Kファイルになります。

マットとは互換性がありません。

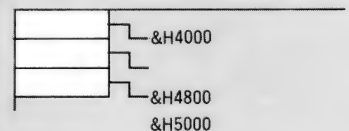
* * *

今回は、特集がメインなのか連載がメインなのかわからなくなってしまいました。ただ、Zに関してはマウス・画像取り込みなど手つかずになってしまったのが残念です。CCHANGEなどはちょっとした変更でまだまだ拡張できるプログラムなので、やりたいことを自分でプログラミングしてみましょう。

図2 X1のG-RAM構成



X1 turbo (320×200) のG-RAM構成は上記のような12枚構造です（ノーマルX1は6枚のみ）。それぞれのG-RAMの左上のアドレスは、&H4000,&H4400,&H8000,&H8400,&HC000で、7から12は裏バンクになっています。



G-RAMの左上を拡大したもので、縦に8段あって、それぞれ&H800ごとのアドレスに割りふられています。

X1turboZでも同じような構成になっていますが、4096色の場合はG-RAM12枚をまとめて2¹²=4096を表しています。青はG-RAMの1,2,7,8・赤は3,4,9,10・緑は5,6,11,12です。BRGそれぞれについて4ビット16階調、かたや8色は1ビット1階調（あるか、ないかだけ）です。

バッファの設定の仕方

グラフィック用バッファを確保する場合もINTEGRAL.Xを書き換えます。

```

1) Z-BASICの場合
I210 MEM$(S_FF,2)=MKI$(&H5000):
MEM$(S_BUFF,2)=MKI$(&H6000)
I220 MEM$(S_BSIZ,2)=MKI$(&H1000):
POKE S-IOMM,4
に差し換えてください。
2) バンクメモリがない機種すべて
I210 MEM$(S_FF,2)=MKI$(&H2000):
MEM$(S_BUFF,2)=MKI$(&H3000)
I220 MEM$(S_BSIZ,2)=MKI$(&H1000):

```

POKE S-IOMM,1

KAME-DOS上からCCHANGE,GLOAD,GSAAVEを使う場合は上記のようにする必要があります。さらに、ノーマルX1の場合は次の1行も付け加えてください。

I225 POKE &HE139,8

2)の書き換えを行ったINTEGRAL.Xでは、CCHANGE,GLOAD,GSAAVEの立ち上げ以外は行わないようにしてください。「DIR」などをすると、画面がメチャクチャになってしまいます。もとへ戻すには、書き換えを行う前のINTEGRAL.Xを起動しなおしてください。

C0E9 7C	138	LD A,H
C0E1 B5	139	OR L
C0E2 28 0A	140	JR Z,SKTPLD
C0E4 FD 7E 00	141	LD A,(IV+0) ;LEFT-DOWN
C0E7 87	142	ADD A,A
C0C8 CB 27	143	ADD A,(IX-3) ;LC(R,N,X+1)
C0EB DD 77 FD	144	LD (IX-3),A
C0EE	145	SKTPLD
C0EE	146	
C0EE FD 7F 00	147	LD A,(IV+0) ;RIGHT-DOWN
C0F0 DD 6E 03	148	ADD A,(IX+3) ;LC(R,N,X+1)
C0F4 DD 77 03	149	LD (IX+3),A
C0F7	150	
C0F7 FD 7E 00	151	LD A,(IV+0) ;RIGHT
C0FA CB 27	152	SLL A
C0FC CB 27	153	SLL A
C0FE FD 77 00	154	LD (IV+0),A
C101 C9	155	RET
C102	156	
C102	157	
C102	158	
C102	159	POINT ;OUT D
C102 CD 9E C1	160	CALL SETHLE
C105 16 00	161	LD D,0
C107 CD 25 C1	162	CALL BCTD02
C10A 01 D0 1F	163	LD BC,;1FDD
C10D ED 78	164	IN A,(C)
C10F F6 10	165	OR \$10
C111 ED 79	166	OUT (C),A
C113 CD 25 C1	167	CALL BCTD02
C116 01 D0 1F	168	LD BC,;1FDD
C119 ED 78	169	IN A,(C)
C11B E6 EF	170	AND \$EF
C11D ED 79	171	OUT (C),A
C11F 7A	172	LD A,D
C120	173	RLCA
C121 07	174	RLCA
C122 07	175	RLCA
C123 57	176	LD D,A
C124 C9	177	RET
C125	178	
C125	179	BCTD02
C125 E5	180	PUSH HL
C126 CD 32 C1	181	CALL BCTD0
C129 01 00 04	182	LD BC,;400
C12C 09	183	ADD HL,BC
C12D CD 32 C1	184	CALL BCTD0
C130 E1	185	POP HL
C131 C9	186	RET
C132	187	
C132	188	BCTD0
C132 4D	189	LD C,L
C133 44	190	LD B,H
C134 ED 78	191	IN A,(C)
C136 A3	192	AND A
C137 28 04	193	JR Z,NOBIT
C139 37	194	SCF
C13A CB 12	195	RL D
C13C C9	196	RET
C13D	197	NOBIT
C13D 87	198	OR A
C13E CB 12	199	RL D
C140 C9	200	RET
C141	201	
C141	202	
C141	203	
C141	204	PSET ;IN D
C141 CD 9E C1	205	CALL SETHLE
C144 CD 5D C1	206	CALL DT0BC2
C147 01 D0 1F	207	LD RC,;1FDD
C14A ED 78	208	IN A,(C)
C14E F6 10	209	OR \$10
C14E ED 79	210	OUT (C),A
C150 CD 5D C1	211	CALL DT0BC2
C153 01 D0 1F	212	LD RC,;1FDD
C156 ED 78	213	IN A,(C)
C15E F6 EF	214	AND \$FF
C15A ED 79	215	OUT (C),A
C15C C9	216	RET
C15D	217	
C15D	218	DT0BC2
C15E E5	219	PUSH HL
C15E CD 6A C1	220	CALL DT0BC
C161 01 00 04	221	LD BC,;400
C164 09	222	ADD HL,BC
C165 CD 6A C1	223	CALL DT0BC
C168 E1	224	POP HL
C169 C9	225	RET
C16A	226	
C16A	227	DT0BC
C16A 4D	228	LD C,L
C16B 44	229	LD B,H
C16C CB 1A	230	RR D
C16E 38 0A	231	JR C,BITON
C170 DE	232	PUSH DE
C171 7B	233	LD A,E
C172 2F	234	CPL
C173 ED 58	235	IN E,(C)
C175 A3	236	AND A
C176 ED 79	237	OUT (C),A
C178 D1	238	POP DE
C179 C9	239	RET
C17A	240	BITON
C17A ED 78	241	IN A,(C)
C17C B3	242	OR E
C17D ED 79	243	OUT (C),A
C17F C9	244	RET
C180	245	
C180	246	
C180	247	
C180	248	XYADR ;IN X,Y OUT ADR,BIT
C180 2A 03 C0	249	LD HL,(X)
C183 ED 5B 05 C0	250	LD DE,(Y)
C187 AF	251	XOR A
C189 32 F6 FB	252	LD (#SCRNM2),A
C18B 06 1D	253	LD B,\$1D
C18D ED 41	254	OUT (C),B
C18F CD 07 59	255	CALL #GRAADR
C192 06 1E	256	LD B,\$1E
C194 ED 41	257	OUT (C),B
C196 22 13 C0	258	LD (ADR),HL
C199 7A	259	LD A,D
C19A 32 15 C0	260	LD (BIT),A
C19D C9	261	RET
C19E	262	
C19E	263	
C19E	264	SETHLE ;OUT HL,E
C19E 3A 12 C0	265	LD A,(ROBP)
C1A1 2A 13 C0	266	

XROT0.X

Watanabe Shinya

渡辺 伸也

皆さんこんばんは。拡大縮小回転という
と、現在のビデオゲームを語るうえでのひ
とつのキーワードになっていますね。僕も
アフターバーナーに感動してからパソコン
でもこういうことができないかなーと思い
始めました。

アフターバーナー以前にもA-JAXとい
うものがあったようですが、僕がゲーセン
に顔を出すようになったのはアフターバー
ナーの出る少し前あたりからなのでA-
JAXの存在すらX68000への移植の話が持
ち上がるまで知りませんでした（ちなみに
それ以前に最後にゲーセンに行ったときは
たしかクレイジークライマーとかがあって、
任天堂のゲームウォッチが流行り始めてい
た頃だったような）。

アフターバーナーの出た頃といえば2年
以上前の話。そんな長いことかかってこの
プログラムを作っていたわけではもちろん
ありません。このプログラムの原形は1年
ほど前にすでにありましたが、「スピードを
追求するあまり、画像がやたら汚い」、「エ
ラーチェックをしないので、ちょっと使い
方を間違っただけですぐにバスエラーが出る
（今回投稿したこのプログラムではさらに
パワーアップしていてハングアップする危
険すらある）」などいろいろなアラがあって
投稿作品としては失格だと考え、投稿は断
念したのでした。

その少しあとにGROT.Xを目にして、
「この分だと近いうちに誰かがプログラム
を発表してX68000ユーザーにとって回転
アルゴリズムは一般的なものになるであろ
う」と読んでいました。が、そうなる気配
はない。アフターバーナーのX68000版が出
たとき、かねがね気になっていた回転プロ
グラムを調べてみますと（電波さんにケチ
をつけるつもりはありませんが）僕が開発
していた過程のアルゴリズムではありませ
んか。

そんなわけで「こういう性格の作品が受
け入れられるかどうか一度Oh!Xの読者&
編集部挑戦してみるか」と考えまして、

今回の投稿へと至ったのです。

注意事項

何度もいいますがこのプログラムはスピ
ードだけがウリで、絵はボロボロ。これで
各種エラーチェック機能をつけて遅くなろ
うものならどーにもこーにも救いなくな
るので、エラーおよび不都合な動作に関す
る責任はどうしてもユーザーに負ってもら
うことになります。

けど、ユーザー側に責任を負わせるソフ
トは悪いソフトだなんていえませんよね。
グラフィックツールや各種言語、特にアセ
ンブラ。よっぽどタコなソフトでなければ
作品の不出来をソフトのせいにはしません
ね。

マシン語なんか暴走するのが常（？）
だから、誰もがテストランする前にRAM
ディスクの内容をセーブするし、大事なフ
ァイルの入ったディスクを実験に使ったり
はしません。これはアセンブラに関しては
「不都合が起こった場合の責任は自分にあ
る」という認識が一般に広まっているから
なのです。暴走したとき、真っ先に考える
のは「自分のミス」であって、「こんなはず
はない！ アセンブラがバグっているの
だ」なんてチラっとでも考えないはずです。

僕としてはその辺が不安の材料なわけで、
「暴走するグラフィック関数」なんて皆さ
んはいままで見たことも聞いたこともない
と思います。でもこのプログラムがそうな
んです。くれぐれも注意してください。こ
れのせいで大事なファイルが消えてしまっ
たなんてことがないように。取り越し苦労
でしょうか。

遊び方

リストはできるだけ多くの方が打ち込む
気になってくれるように短くしたつもりで
す。まず、リストを入力します。コンパイ
ルは、

読者投稿による画面回転プログラムです。
比較的小さなリストでも効果てきめん。256
色の画像をグルグル回します。特殊効果そ
の他、画像処理の際に参考にしてください。
なお、高速化のためエラー処理など一部省
略された処理がありますので注意してくだ
さい。



CC /W XMKDAT0.C

です。そして実行ですが、このプログラム
はカレントディレクトリに約30Kバイトの
ファイルを作成しますので、カレントには
その分の余裕が必要です。

プログラムを実行すると放射線が描かれ、
中心が抜けていきます。放射線は360度制で
3度おきに120本。抜けた中心部分は直線デ
ータとして、ファイルに吸収したのです。
そしてそのファイルが出力されるXROT
DAT0です。

あとはXROT0.SとTESTROT.Cを入力
して、

CC /W /Y /w TESTROT.C
XROT0.S

とすればTESTROT.Xができます。

実行時にはXROT0.Xがカレントにあ
るようにしてください。また、実行する前
に16色または256色モードの絵をページ0
にロードしておきます。絵は512×512ドッ
トいっぱい描き込んであるものを選んで
ください（65536色のデータはto256.xなど
で変換してください）。

適当なデータがないときはTESTROT.
Cのコメント化してある行を有効にしま
しょう。操作方法は図1ですが、まずは
こちらの指示に従って操作してください。
では実行です。

TESTROT

まずOPT.1キーで、縮小していきます。
するとすぐに画面はページ0の大きさを越
え、本来の画像の周囲に変な画像が出現し
ます。そのあたりで操作を止めて、ページ

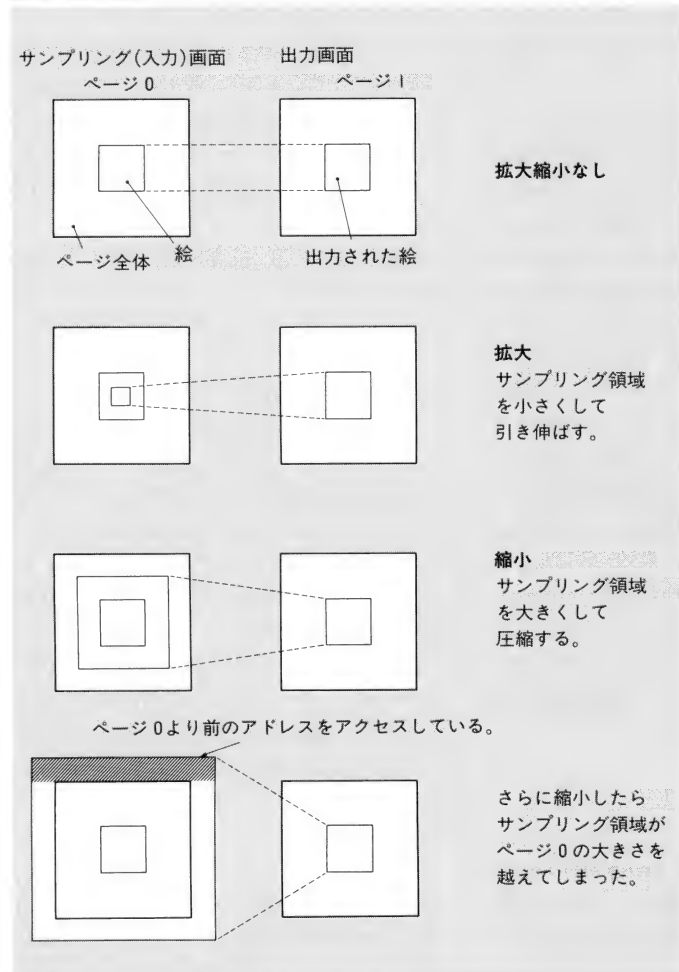
0の大きさを越えた画面上部の黒い部分に注目してください。拡大縮小のアルゴリズムは図2です。

この黒い部分はページ0のアドレス(\$C00000)より前の領域からデータを読んでいることがわかります。しかしRAMをフル装備したマシンでない限りこの領域にメモリは存在しないわけで、普通この領域をアクセスすれば「バスエラーが発生しました」となるのですが、いまそうならないのは画像を作成しているあいだだけ、バスエラーベクタを書き換えてオリジナルのバスエラー例外処理プログラムで処理しているからなのです。このへんは実際にリストを打ち込んだ方なら察しがついていると思います。

ほとんどのユーザーのマシンでこの黒い部分の面積に比例して処理が遅くなりますが、これはバスエラーの数だけ例外処理にとんでいるためです。

絵が左右に連なっているのも含めて、このプログラムでは「絵からはみ出したかどうかチェックして回避しない」のが諸悪の根源なわけですが、その処理を入れると極

図2 拡大縮小



端に遅くなってしまうのです。

ところで画面の下の方に見える「変な画像」についてですが、まずTESTROT.Xの動作を理解してください。図3です。

縮小するとサンプリング領域が広まってページ0の上をサンプリングしてバスエラーを出しますが、ページ0の下もサンプリングします。ページ0の下とはページ1、

図1 TESTROT.Xのキーボード操作

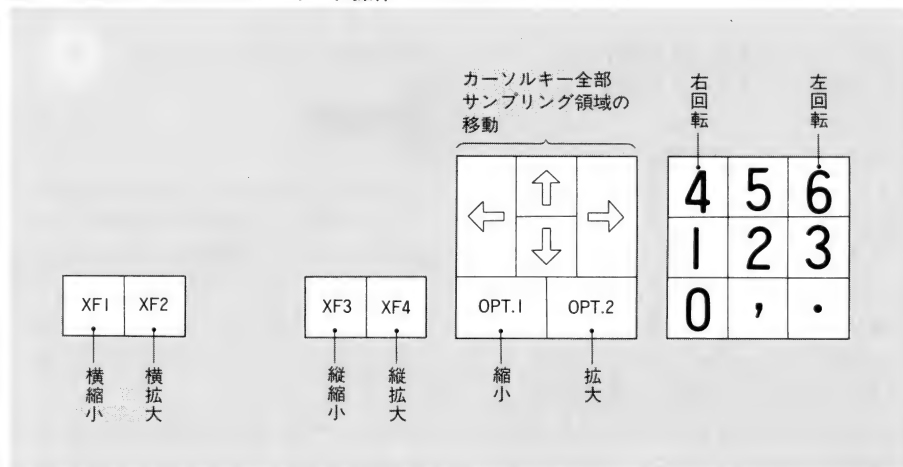
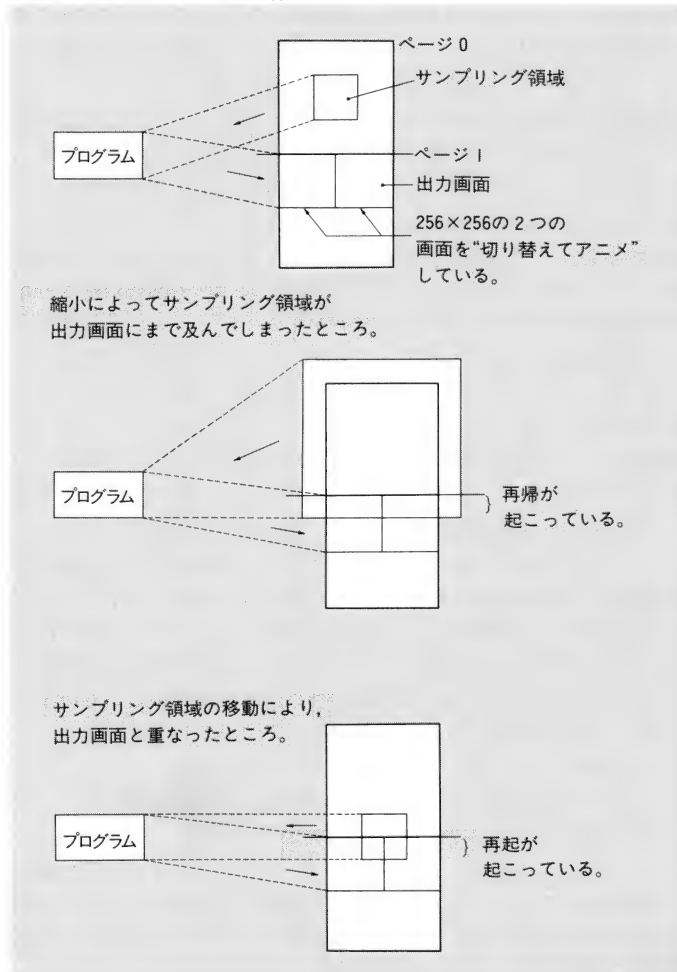


図3 TESTROT.Xの動作



ひと通りいじくって使い勝手をのみこんだらXROT0.Sのバスエラー回避部分を切ってしまうでしょう。さすがにこれは危なすぎるので。描画処理中に割り込んできたプログラムがバスエラーを起こした場合にも気まずいものがあるし。あと、このバスエラー回避は読み込み側画像のエラー処理専門なので、それ以外のバスエラーだとハングアップします。

以後、その絵やページからはみ出すようなパラメータの設定をしないように気をつけましょう。

関数の説明

XROT0はいうまでもなくXC対応に作られた関数で、数Kバイトのプログラム部と、約30Kバイトのデータ部に分けられます。実際にもデータ部はXROTDAT0として別個に存在していて、プログラムが立ち上がったあとに読み込みます。本来プログラムとデータは一体化していました。これはほかのOS上（自作とか）でも動かすことを想定していたからで、Humanのコマンド（この場合はディスク関係）は使いたくなかったのです。しかしそれだとリンクが長いので今回の投稿では作り直しています。

なおXROTDAT0のあるディレクトリはXROT0.Sのデータ文で決定されるので、ここを書き換えてしまえば、どこにファイルがあろうとかまいません。

XROT0は以下の3つの関数から成り立っています。

XROT0INIT();

ディスクからデータを読み込みます。プログラムが立ち上がったなら、1回実行してください。実行しなかったら素直にバグるのみです。純正のグラフィックコマンドは「画面初期化コマンドを実行していない」ことを察知するとシカトしてくれますね。IOCSレベルからしてそういう構造になっていますが皆さんはこういうのを親切な設計だと思いますか？

IOCSといえばあのレジスタをピシバシ使うやり方はいただけません。一度作った値をメモリに待避しないでそのままレジスタに残しておいて使えるのが68000のプログラミングスタイルであり、68000の価値だと考えます。Cのようにパラメータをスタックで渡すとか、パラメータ群のセットしてあるメモリの先頭ポインタをスタックで渡すとかのほうがよいと思うのですがどうでしょう。アセンブラで組むときはまずスーパーバイザモードにして、I/Oポートの

操作は自力でやる人って多いと思います。

WNDROT0(P0,P1);

INT P0;

入力画像のあるページ番号（0～3）

INT P1;

画像を出力するページ番号（0～3）

P0=P1であってもかまわない。負数禁止。

実際にプログラムがほしがっているのは各ページ番号ではなくて、各ページの座標（0,0）のアドレスです。それをP0,P1より計算して内部に控えておきます。計算式は、

アドレス=\$C00000+\$80000×ページ番号

です。0～3以外の値も受け付けます。たとえば4だと\$E00000、つまりテキストVRAMを指定できます。XROT0は拡大縮小回転しなければ、ただの画像転送命令として使えるので、たとえば、

WNDROT0(0,4);

としてテキストVRAMに絵を置いておき、以後、

WNDROT0(4,0);

とすれば、グラフィックVRAMに入力画像を置いておく必要はなくなります。

ただし、XROT0はメモリのどこを指定しても実画面512×512ドットのフォーマット（X方向のカウントが±2バイト、Y方向のカウントが±1024バイト）として扱うので、16色モードの絵を1ページ分転送してもテキストVRAMの全部を使ってしまうます。

XROT0(X1,Y1,X2,Y2,W,H,SX,SY,A);

int X1: 入力画像の中心のX座標

int Y1: 入力画像の中心のY座標

int X2: 出力画像の中心のX座標

int Y2: 出力画像の中心のY座標

int W: 出力画像の横サイズの1/2

int H: 出力画像の縦サイズの1/2

int SX: 横の拡大縮小率（下位2バイトのみ有効）

int SY: 縦の拡大縮小率（下位2バイトのみ有効）

int A: 回転角度 ±3900000

中心の座標とはいわずと知れたその画像の縦と横の中心に位置するドットの座標ですが、中心の1ドットが存在するためには画像のサイズが縦横ともに奇数でなければなりません。

しかるに画像のサイズは1/2の状態で指定するので、サイズは必ず偶数になり真の中心ドットはなくて代わりに、候補の4ドットが存在するかたちになります。ではどうするかというと、4ドットのうちの任

意の1ドットを中心として決めてしまってもよいのです。どうせ精度はガタガタなので1～3ドットの違いなど問題になりません。

ですが拡大縮小なしで回転角度が0, 30, 60, 90, 120の倍数のときはさすがに正確に動きますのでそのあたりも考慮してください。

サイズを1/2で指定させるのはこの値で計算することが多いのと、1/1の値から1/2の値を作ろうとすると、奇数だった場合に誤差が出てしまうからです（整数演算なので）。精度はガタガタだと書きましたが、なるべくそうならないようにはしているのです。

拡大縮小率の設定はややこしくて、任意の拡大縮小率の逆数を16進数の固定小数点小数として考えます。小数点は下位2バイトのあいだにとって、\$00.00とします。

概念としては、

2倍拡大 → 1/2 → \$01.00/2 → \$00.80 → \$0080

1/2縮小 → 2/1 → \$01.00 * 2 → \$02.00 → \$0200

となります。

実用的な計算法は、

double A = 1.25;

/*実数によるわかりやすい表記*/

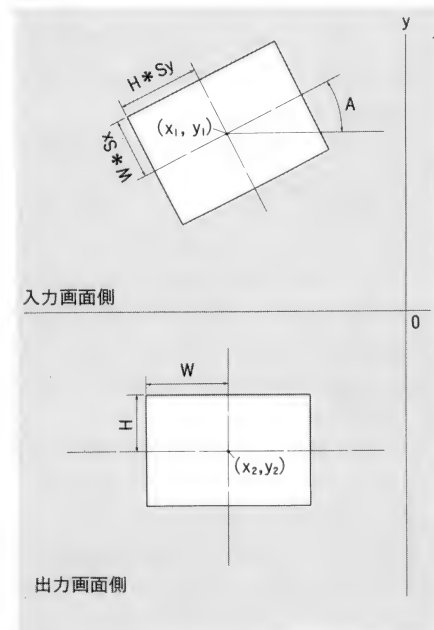
int SX,SY;

SX = (int)(1.0 / A * 256.0);

です。

なお、いまの状態では縮小率は1/4までですが、この制限を加えているのは、XROT0.Sの18行と19行だけですので変更するのは簡単です。回転角度の制限はDIVSによる

図4



割り算の限界のことです。

回転のアルゴリズム

いろいろな回転プログラムが出まわっていますが、どれも似たような作りをしています。僕は非常に奥が深いものかと思ってさまざまな試みをしました。その結果、誰もが最初に思いつくアルゴリズムが実は一番優秀であることがわかってきました。なんか残念です。

簡単に説明すると、

「目的の画像より水平に1ライン分のデータをサンプリングし、それに回転処理を施し目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

「目的の画像より回転処理を施したライン上（つまり水平垂直を含むナナメ線）に1ライン分サンプリングし、それを水平に変換して目的のVRAMに出力する。これを縦の大きさだけ繰り返す」

わけです。前者を画像回転型、後者を座標回転型と僕は呼んでいます。

この投稿は後者の座標回転型であり、描画面積が一定なため「描画速度が一定で次の絵を出すとき、前の絵をクリアすることなくそのまま書きしこめる」というアニメ処理に好都合な特徴を持っています。

画像回転型は描画面積が不定なので、同じ画像データでも縮小しているほど速く処理が終わるような作り方が可能です。あるいは工夫が足りないと、拡大縮小によって処理速度が違ってくるような作りになってしまいがちです。またアニメ処理のときはいちいち前の絵をクリアする必要があります。こう書くと座標回転型に比べて悪いことづくめのようなようですが、小規模な処理に向いているのでゲームにはこちらのほうがよく採用されるようです。

2つのアルゴリズムは実はただ単にデータの流れが逆になっているだけだ、ということに気がつかれましたか？ この2つは基本的には同じひとつのアルゴリズムです。

いわれてみれば簡単な構造でアルゴリズムのうちに入らないかもしれませんが。回転処理の難しさはプログラムそのものよりも処理の概念やイメージをつかみにくいところにあるように思います。

スピードアップのポイントは1ラインの転送をいかに速くするかにかかっており、転送は普通ループで処理しているのでこれを展開します。展開すると画面の横幅が固定されますが、このプログラムでは自己書き換えによって1ライン分の転送ルーチン

を「作る」ので可変長になっています。

また、この転送ルーチンの中で、

LEA d16(An),An

が使われています。これは最高速の32ビット加減算命令なのですが、イミディエイトであるd16の部分はメインメモリ上の値なのでこれを変更したくば、またもや自己書き換えです。よってこのプログラムではひとつの領域に2カ所から書き換え動作を行うことによって、ひとつの転送ルーチンを作り出しています。

XROT0は1ドット/20クロックの描画速度を持っているので、256×256ドットの画像だと秒速7コマで書き換えることがで

きます。いろいろとサイズを変えて実験してみましょう。

あと注意が必要なのはVPAGE、HOMEなどの関数で、これらは垂直帰線期間を無視して動くので、垂直帰線期間待ちをする処理が別個に必要だということです。TES TROT.Cでは#asmでやっている部分です。Cで作ることもできますが、アセンブラで2行だと知っているのとCを使う気になりません。

XROT0の拡大回転処理はウソ臭いですがね。本当だったらOh!FM 3月号の「view.exp」のように拡大した四角い1ドットにも回転処理を加えなければならないところ

回転について

通常の座標系の上に回転させた座標系を想定し、その座標系上の一定領域（長方形）の中の座標を1ドットずつ順番に指定できるシステムを作る。

通常の座標系上の一定領域（長方形）の中の座標を1ドットずつ順番に指定できるシステムも作る。

そしてこの2つを同時に動かす。このときデータの流れ（受け渡し）が、

通常座標→回転座標のとき画像回転型

回転座標→通常座標のとき座標回転型となる。

回転座標を想定しても、ドットの並びはあく

までも通常座標の方眼なので、そこに大きな無理が生ずる。それはドット画面に真の斜線が描けないのと同じである（階段になってしまう）。したがってドット構成の画面である限り、真の画像回転は不可能であり、すべて疑念的なものになる。ハードウェアによる回転でもその例にみれずチラついている。

回転処理に使う画像データはなるべくチラつきを目立たなくするためにグラデーションを多用してボカシ気味に描くのがコツである。ゲーセンに行ったら確かめてみよう。本当に綺麗に回転させたくば何千色も使って色の補間をする処理が必要である。

リスト1

```
===== XMKDAT0.C =====
1: #include "basic0.h"
2: #include "BASIC.h"
3: #include "graph.h"
4: #include "math.h"
5: #include "stdio.h"
6:
7: main()
8: {
9:     int a,b,c,sampx,sampy,ax,ay;
10:    short int x[128];
11:    double DEG,SC,CO;
12:    FILE *fi;
13:    DEG = pi() / 180.0;
14:    screen( 2, 0, 1, 1 );
15:    home( 0, 128, 128 );
16:    window( 0, 0, 1023, 1023 );
17:
18:    fi = fopen( "XROTDAT0", "wb" );
19:    for( a = 0; a < 360; a += 3 ){
20:        ax = (int)( 370.0 * cos( (double)a * DEG ) );
21:        ay = (int)( 370.0 * sin( (double)a * DEG ) );
22:        line( 512, 512, 512 + ax, 512 - ay, 10, 'NASI' );
23:        sampx = 512; sampy = 512;
24:        c = a / 90; c = a - c * 90;
25:        if( c > 45 ) c = 90 - c;
26:        CO = 1.;
27:        SC = cos( (double)c * DEG );
28:        for( b = 0; b <= 127; b++ ){
29:            if( point(sampx, sampy - 1) == 10 ){ ax = 0; ay = -1; }
30:            if( point(sampx + 1, sampy - 1) == 10 ){ ax = 1; ay = -1; }
31:            if( point(sampx + 1, sampy ) == 10 ){ ax = 1; ay = 0; }
32:            if( point(sampx + 1, sampy + 1) == 10 ){ ax = 1; ay = 1; }
33:            if( point(sampx, sampy + 1) == 10 ){ ax = 0; ay = 1; }
34:            if( point(sampx - 1, sampy + 1) == 10 ){ ax = -1; ay = 1; }
35:            if( point(sampx - 1, sampy ) == 10 ){ ax = -1; ay = 0; }
36:            if( point(sampx - 1, sampy - 1) == 10 ){ ax = -1; ay = -1; }
37:
38:            CO = CO + SC;
39:
40:            if( CO > 1. ) {
41:                CO = CO - 1.;
42:                pset(sampx,sampy,0);
43:                sampx += ax; sampy += ay;
            }
```


をXROT0では単にソフト的にドットを粗くしただけだったりします。

ところで、このウソ臭い回転、あのアフターバーナー（もちろん本物）がやっているのを知っていますか？ ここからは僕の憶測ですが、アフターバーナーのハードはそれまでのセガの体感シリーズであるスペハリ、エンデューロレーサー（マイナー）と同じで、スプライトには拡大縮小機能しかありませんでした。

アフターバーナーは2MバイトのRAM（と聞いた）を増設し、そこに回転パターンをこさえてパターン持ちの回転処理をするというパソコンライクな作りをしていたのです。

改造のポイント

XROT0では画像を小さく設定すれば当然処理が高速になります。が、もうひとつ高速化する方法があります。それはソースリストを書き換えることになりませんが、XROT0.Sの、

197行を有効にする

200行を有効にする

201行を無効にする（コメント化する）

210行を有効にする

ことです。どうです？ なかなかうまいことやっただと思いませんか？

しかし本当はプログラムを皆さんが理解して、そのうえで自力で改造していただくのが理想です。ですが本気でプログラムを解説すると何ページあっても足りないのもそれは諦めました。

その他諸々

回転というとすぐに「アサルト！」とか「ダートフォックス（メタルホークでない）ところがポイント。このゲーム好きなんですけど廃れるのが早いのです。しくしく。CD買いました）の移植だ！」とか聞こえてきそうですが、それは無理というものです。

処理速度の問題もありますがここを強調したいのです。XROT0は「1枚絵の回転」ですが、ナムコのシステム2は「BG画面の回転」です。BGとはX68000に搭載のあのスプライトBGのことです。ですからまるっきり違うのだということを理解してください。多くの人は回転ばかりに気を取られているようですけど。

しかし自分でプログラムを組もうとでもしてみない限りそこまで考えないのは当然

```
44:         }else{
45:             ax = ay = 0;
46:         }
47:         x[b] = 2 * ax + 1024 * ay;
48:     }
49:
50:     fwrite( (char *)&x, 2, 128, fi );
51:     printf( "%d\n", x[127] );
52: }
53:
54: for( a = 0; a <= 359; a += 3 ){
55:     putw( ( short int )(sin( a * DEG ) * 4096.0 ), fi );
56:     putw( ( short int )(cos( a * DEG ) * 4096.0 ), fi );
57: }
58: fclose( fi );
59: screen( 2, 0, 1, 1 );
60: }
```

リスト2

```
===== TESTROT.C =====
1: #include "basic0.h"
2: #include "graph.h"
3: #include "doslib.h"
4: #include "iocslib.h"
5:
6: main()
7: {
8:     int a,b,c,d,SSP;
9:     int X1,Y1,X2,Y2,W,H,SX,SY,A;
10:    int a_frag;
11:
12:    SSP = B_SUPER(0);
13:    C_CUROFF(); A_CLR_AL();
14:    CRTMOD( 10 + 0x100 );
15:
16:    /***** S A M P L E *****/
17:    screen( 0, 2, 1, 1 );
18:    window(0,0,511,511);
19:    apage(0);
20:    for(a = 0; a < 16; a++ ) palet( a, rgb( 2*a, 2*a, 2*a ) );
21:
22:    fill( 0, 0, 511, 511, 15 );
23:    fill( 10, 10, 501, 501, 0 );
24:
25:    for(a = 0; a < 511; a += 64){
26:        for(b = 0; b < 511; b += 64){
27:            for(d = 0; d < 23; d += 4){
28:                c = ( a + b + c ) - ( ( a + b + c ) / 15 ) * 15;
29:                fill( a+d, b+d, a+50-d, b+50-d, c );
30:            }
31:        }
32:    }
33:    box( 256-128, 256-128, 256+128, 256+128, 15, 'NASI' );
34:    fill( 256-40, 256-40, 256+40, 256+40, 15 );
35:    fill( 256-30, 200-30, 256+30, 200+30, 10 );
36:    *****/
37:
38:    vpage(2);
39:    SX = SY = 256;
40:    X1 = Y1 = 256;
41:    Y2 = 128;
42:    W = 128;
43:    H = 128;
44:    A = 0;
45:    a_frag = 1;
46:    XROT0_INIT();
47:    WNDROT0(0,1);
48:
49:    while(1){
50:        if( BITSNS( 0x8 ) & 0x80 ) A -= 1; /* 4 key */
51:        if( BITSNS( 0x9 ) & 0x02 ) A += 1; /* 6 key */
52:
53:        if( BITSNS( 0xE ) & 0x04 ) { SX += 10; SY += 10; } /* OP1 key */
54:        if( BITSNS( 0xE ) & 0x08 ) { SX -= 10; SY -= 10; } /* OP2 key */
55:
56:        if( BITSNS( 0xA ) & 0x20 ) SX += 10; /* XF1 key */
57:        if( BITSNS( 0xA ) & 0x40 ) SX -= 10; /* XF2 key */
58:
59:        if( BITSNS( 0xA ) & 0x80 ) SY += 10; /* XF3 key */
60:        if( BITSNS( 0xB ) & 0x01 ) SY -= 10; /* XF4 key */
61:
62:        if( BITSNS( 0x7 ) & 0x08 ) X1 -= 5; /* LEFT key */
63:        if( BITSNS( 0x7 ) & 0x20 ) X1 += 5; /* RIGHT key */
64:
65:        if( BITSNS( 0x7 ) & 0x10 ) Y1 -= 5; /* UP key */
66:        if( BITSNS( 0x7 ) & 0x40 ) Y1 += 5; /* DOWN key */
67:
68:        if( BITSNS( 0x0 ) & 0x02 ) break; /* ESC key */
69:
70:        #asm
71:
72:        VDISP: BTST.B #4,$E88001 /* 帰線待ち */
```


です。偉ぶった文章ですがそういうつもりはありません。

なんか悲観的になりましたが「このプログラムでは無理だ」という話です。ふたたび誤解のないようお願いします。もちろん僕はBG回転に挑戦するつもりです(図5)。

BG回転機能を持ったハードを販売しているのはいまのところナムコだけとされます。そのアーケードマシンでもロクにない機能を家庭用ゲームマシンに持ち込もうというのだから、(よくも悪くも)いかにとんでもないことをスーパーファミコンがやろうとしているかがわかると思います。コストが下げられなくて当然、発売が延びて当たり前といえますね。

* * *

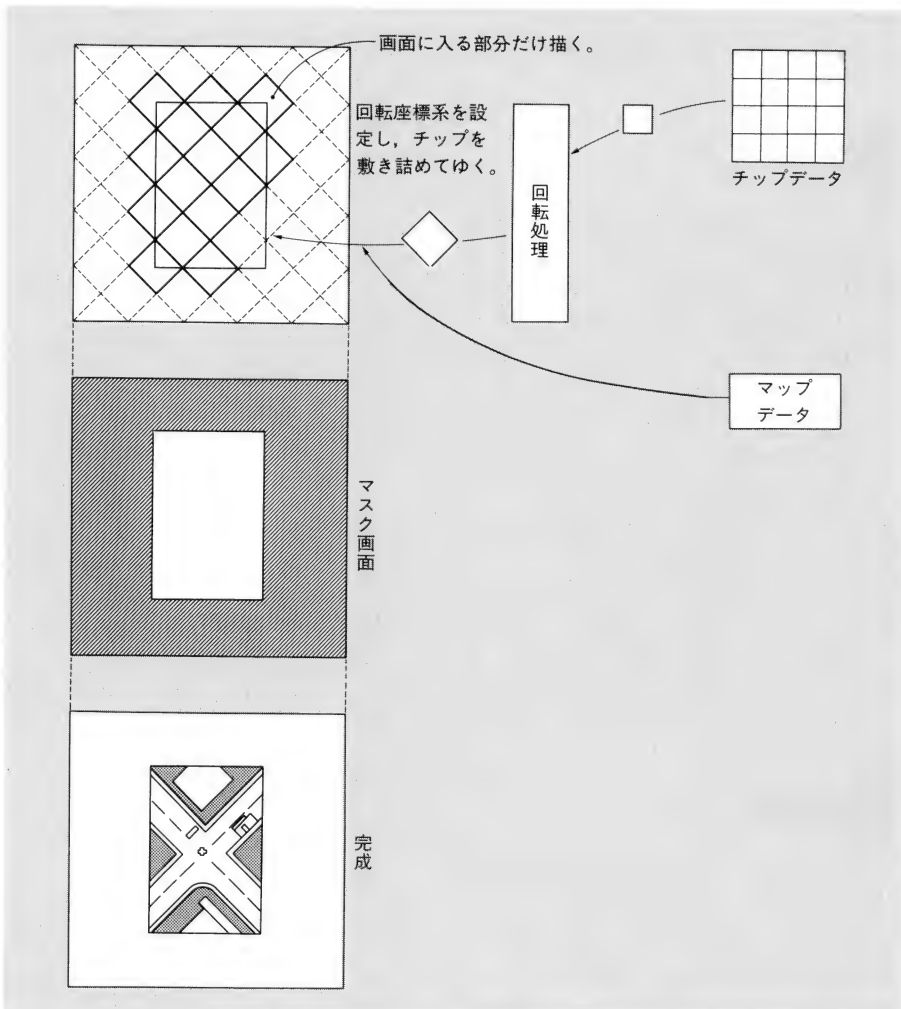
XROT0ということはXROT1があるだろうと容易に想像がつくわけですね。XROT1はXROT0の20%の処理速度向上を果たしたものの、画面のサイズが128ドットまでに限られるのと仮想画面の使用を強要されるという、面白くない副作用がぞくぞくと発生したので、発表はXROT0にさせていただきました。XROT1が出ないのなら識別のために「XROT」に「0」をつけておく必要はないのですが、これは単に僕の気持ちの問題です。

おしまい

いかがでしたか? このプログラムでX68000の限界のひとつを示したという自信があります。柴田惇氏のいい回しを借りれば「すごい」と思おうが「こんなもんか」と思おうが今後のX68000ユーザーの指針になることは確かである、というところでしょうか。あと僕としては「その筋」は死語になってほしくないで、みんなで盛り返しましょう。いろいろ偉そうなことを書いてきましたが、僕はOh!X誌上ではあまりでしゃばれないような気がします。いまさら遅いか。ではさようなら。

```
73:                                BNE.B    VDISP
74:
75:                                #endasm
76:
77:                                if( a_frag == 1 ) { X2 = 128      ; home(1,256,0); }
78:                                else { X2 = 128 + 256; home(1,000,0); }
79:
80:                                a_frag *= -1;
81:
82:                                XROT0( X1, Y1, X2, Y2, W, H, SX, SY, A );
83:
84:                                }
85:                                CRTMOD( 8 + 0x100 );
86:                                C_CURON();VPAGE(1);
87:                                B_SUPER( SSP );
88: }
```

図5 回転BGシステム



リスト3

```
===== XROT0.S =====
1: .GLOBL _XROT0_GO
2: .GLOBL _XROT0_INIT
3: .GLOBL _WDR0T0
4:
5: .TEXT
6:
7: ***** ROT 0 *****
8: * SV MODE *****
9: _XROT0: MOVEM.L D0-D7/A0-A4,REGBUF
10: MOVEM.L #81,D0 ; _B_SUPER
11: MOVEM.L #0,A1 ;
12: TRAP #15 ;
13: MOVEM.L D0,SSPBUF ;
14: MOVEM.L USP,A0 ;
15: MOVEM.L A0,USPBUF ;
16:
17: * ERROR CHECK *****
18: AND.W #03FF,30(SP) ;
19: AND.W #03FF,34(SP) ;
20: CMPI.W #128,26(SP) ; IF 128 < H THEN ERROR
21: BHI.W BAD_END ;
```

```
22: MOVEM.W 22(SP),D0 ;
23: BEQ.W BAD_END ; IF W = 0 THEN ERROR
24: CMPI.W #128,D0 ; IF 128 < W THEN ERROR
25: BHI.W BAD_END ;
26: CMPI.W NOW_WIDE,D0 ; W は前回の設定値と同じか
27: BEQ.B GO ; 同じなら書き換えしない。
28: MOVEM.W D0,NOW_WIDE ;
29:
30: *書き換えサブルーチン*****
31: LEA.L W_LINE,A0 ;
32: MOVEM.W 22(SP),D0 ;
33: ASL.W #1,D0 ;
34: SUBQ.W #1,D0 ;
35: LOOP00: MOVEM.W #34D1,(A0)+ ; #34D1 = MOVE.W (A1),(A2)+
36: MOVEM.L #43E9_0000,(A0)+ ; #43E9_0000= LEA.L 0000(A1),A1
37: DBRA.W D0,LOOP00 ;
38: MOVEM.W #4E75,(A0) ; #4E75 = RTS
39:
40: *角度取り出し*****
41: GO: MOVEM.L 36(SP),D0 ; D0 = A
42: MOVEM.L #120,D1 ;
43: DIVS.W D1,D0 ; D0 = -120 ... +119
```



```

44: SWAP.W D0
45: EXT.L D0
46: ADD.L D1,D0
47: DIVU.W D1,D0
48: SWAP.W D0
49: EXT.L D0
50: MOVE.L D0,D6
51:
52: *転送側画像 転送開始座標*****
53: ASL.W #2,D0
54: LEA.L XSDAT0,A0
55: MOVE.L (A0,D0.W),D4
56: MOVEQ.L #12,D7
57: MOVE.W D4,D5
58: SWAP.W D4
59:
60: MOVE.W 30(SP),D0
61: MULU.W 22(SP),D0
62: ASR.L #8,D0
63: NEG.W D0
64: MOVE.W 34(SP),D1
65: MULU.W 26(SP),D1
66: ASR.L #8,D1
67: MOVE.W D5,D2
68: MOVE.W D4,D3
69:
70: MULS.W D0,D2
71: MULS.W D1,D3
72: SUB.L D3,D2
73: ASR.L D7,D2
74:
75: MULS.W D5,D1
76: MULS.W D4,D0
77: ADD.L D0,D1
78: ASR.L D7,D1
79: MOVEQ.L #10,D7
80:
81: MOVE.W 6(SP),D4
82: MOVE.L 8(SP),D5
83: ADD.W D2,D4
84: SUB.L D1,D5
85: ASL.W #1,D4
86: ASL.L D7,D5
87: MOVEA.L SAMPL,A0
88: ADDA.W D4,A0
89: ADDA.L D5,A0
90:
91: *合成側画像 描画開始座標*****
92: MOVEA.L PLAYPL,A2
93: MOVE.W 14(SP),D0
94: MOVE.L 16(SP),D1
95: SUB.W 22(SP),D0
96: SUB.L 24(SP),D1
97: ASL.W #1,D0
98: ASL.L D7,D1
99: ADDA.W D0,A2
100: ADDA.L D1,A2
101:
102: *使用する直線データ二本の先頭アドレス***
103: LEA.L XROTDAT0,A3
104: MOVEA.L A3,A4
105: MOVE.W D6,D1
106: ASL.W #8,D1
107: ADDA.W D1,A3
108: MOVE.L D6,D1
109: ADDI.W #90,D1
110: DIVU.W #120,D1
111: SWAP.W D1
112: ASL.W #8,D1
113: ADDA.W D1,A4
114:
115: *「書き換えルーチン」のLEAのイミディエイト値を書き換える。*
116: * これは横方向拡大縮小処理
117: MOVE.W 30(SP),D0
118: MOVE.W 22(SP),D2
119: ASL.W #1,D2
120: MOVE.W D2,D7
121: SUBQ.W #1,D2
122: LEA.L W LINE+4,A1
123: CLR.W D3
124: TST.B 30(SP)
125: BNE.B NEXT00
126:
127: *横方向拡大時の書き換え*****
128: LOOP01: CLR.W (A1)
129: SUB.B D0,D1
130: BCC.W NEXT01
131: MOVE.W (A3,D3.W),(A1)
132: ADDQ.B #2,D3
133: ADDQ.W #6,A1
134: DBRA.W D2,LOOP01
135: BRA.B NEXT02
136:
137: *横方向縮小時の書き換え*****
138: NEXT00: MOVE.W D0,D4
139: LSR.W #8,D4
140: SUBQ.W #1,D4
141: LOOP02: CLR.W (A1)
142: MOVE.W D4,D5
143: LOOP03: MOVE.W (A3,D3.W),D6
144: ADD.W D6,(A1)
145: ADDQ.B #2,D3
146: DBRA.W D5,LOOP03
147: SUB.B D0,D1
148: BCC.B NEXT03
149: MOVE.W (A3,D3.W),D6
150: ADD.W D6,(A1)
151: ADDQ.B #2,D3
152: NEXT03: ADDQ.W #6,A1
153: DBRA.W D2,LOOP02
154:
155: *直線データをバッファに移し、拡大縮小処理を施しておく。*
156: * これは縦方向拡大縮小処理
157: NEXT02: MOVE.W 34(SP),D0
158: ASL.W 26(SP)
159: SUBQ.W #1,26(SP)
160: LEA.L LINEY,A3
161: MOVE.W 26(SP),D2
162: CLR.W D3
163: TST.B 34(SP)
164: BNE.B NEXT04
165:

```

```

166: *縦方向拡大時*****
167: LOOP04: CLR.W (A3)
168: SUB.B D0,D1
169: BCC.B NEXT05
170: MOVE.W (A4,D3.W),(A3)
171: ADDQ.B #2,D3
172: NEXT05: ADDQ.W #2,A3
173: DBRA.W D2,LOOP04
174: BRA.B NEXT06
175:
176: *縦方向縮小時*****
177: NEXT04: MOVE.W D0,D4
178: LSR.W #8,D4
179: SUBQ.W #1,D4
180: LOOP05: CLR.W (A3)
181: MOVE.W D4,D5
182: LOOP06: MOVE.W (A4,D3.W),D6
183: ADD.W D6,(A3)
184: ADDQ.B #2,D3
185: DBRA.W D5,LOOP06
186: SUB.B D0,D1
187: BCC.B NEXT07
188: MOVE.W (A4,D3.W),D6
189: ADD.W D6,(A3)
190: ADDQ.B #2,D3
191: NEXT07: ADDQ.W #2,A3
192: DBRA.W D2,LOOP05
193:
194: *合成側画像の改行値*****
195: NEXT06: LEA.L LINEY,A3
196: MOVE.W 26(SP),D0
197: *ASR.W #1,D0
198: ASL.W #1,D7
199: NEG.W D7
200: *ADDI.W #2048,D7
201: ADDI.W #1024,D7
202:
203: *転送開始 *****
204: MOVE.L $0008,BUS_ERROR
205: MOVE.L $CANT,$0008
206:
207: LOOP07: MOVEA.L A0,A1
208: BSR W LINE
209: ADDA.W (A3)+,A0
210: *ADDA.W (A3)+,A0
211: ADDA.W D7,A2
212: DBRA.W D0,LOOP07
213:
214: MOVE.L BUS_ERROR,$0008
215:
216: *S V M O D E E N D*****
217: BAD_END:
218: MOVE.L USPBUFF,A0
219: MOVE.L A0,USP
220: MOVEQ.L #81,D0
221: TST.L SSPBUF
222: BMI NEXT08
223: MOVEA.L SSPBUF,A1
224: TRAP #15
225: NEXT08: MOVEM.L REGBUF,D0-D7/A0-A4
226: RTS
227:
228: *その場液ぎのバスエラー例外処理
229: CANT: LEA 8(SP),SP
230: *LEA $FC0000,A1
231: CLR.W (A2)+
232: RTE
233:
234: *****X R O T O _ I N I T*****
235: XROT0_INIT:
236: MOVEM.L D0-D1,-(SP)
237: MOVE.W #0,-(SP)
238: PEA DATFILE
239: .DC.W $FF3D
240: MOVE.W D0,D1
241:
242: MOVE.L #87800+$1E0,-(SP)
243: PEA XROTDAT0
244: MOVE.W D1,-(SP)
245: .DC.W $FF3F
246:
247: MOVE.W D1,-(SP)
248: .DC.W $FF3E
249:
250: LEA 6+10+2(SP),SP
251: MOVEM.L (SP)+,D0-D1
252: RTS
253:
254: *****X R O T O _ I N I T E N D*****
255: *****W N D R O T _ I N I T*****
256: WNDROT0:
257: MOVE.W 6(SP),D0
258: LEA.L $B80000,A0
259: LOOP08: ADDA.L #80000,A0
260: DBRA.W D0,LOOP08
261: MOVE.L A0,SAMPL
262: MOVE.W 10(SP),D0
263: LEA.L $B80000,A0
264: LOOP09: ADDA.L #80000,A0
265: DBRA.W D0,LOOP09
266: MOVE.L A0,PLAYPL
267: RTS
268:
269: *****W N D R O T _ I N I T E N D*****
270:
271: .DATA
272: NOW_WIDE: .DC.W 0
273: SAMPL: .DC.L $C00000
274: PLAYPL: .DC.L $CC0000
275: DATFILE: .DC.B "XROTDAT0",0
276: .BSS
277: .EVEN
278: USPBUFF: .DS.L 1
279: SSPBUF: .DS.L 1
280: BUS_ERROR: .DS.L 1
281: REGBUF: .DS.L 8+5
282: LINEY: .DS.W 256
283: W LINE: .DS.W 3+256+2
284: XROTDAT0: .DS.B $7800
285: XSDAT0: .DS.B $1E0
286: .END

```


HEART・負けるが勝ち

Ikeya Masahiko 池谷 昌彦

読者投稿によるCARD.FNCを使用したトランプゲームです。「負けるが勝ち」という副題どおり、できるだけカードを取らないようにゲームをすすめなければいけません。なお、このプログラムの実行には1990年5月号で発表されたCARD.FNCが必要です。



私は以前より自分でカードデータを作ってゲームを作っていましたが、数度に分けてPUTするという方法では遅くてBASICでは使いものにならず弱っていました。その点、CARD.FNCはBASICでも十分実用になるので嬉しくなります。さっそく、これを使ってカードゲームを作ってみました。

ゲームの名前はHEARTです。「負けるが勝ち」と副題をつけたいと思います。このゲームは3人から6人用のトランプゲームです。4人でプレイするのがもっともバランスがとれるので、プレイヤーはコンピュータ3人と人間ひとりの構成にしました。

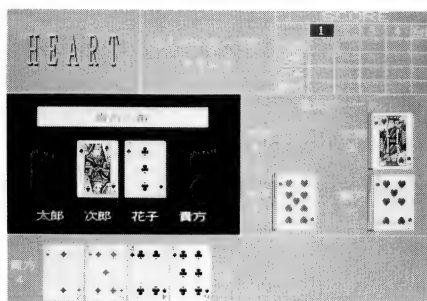


ゲームの内容

ゲームのルールを簡単に説明しましょう。まず、各プレイヤーは13枚ずつ手札を持ちます。親から順番に1枚ずつ手札を場にさらしていきます。このとき出せるのは親が出した台札と同じスート（記号）のカードだけです。どうしても出せない場合はなにを出してもかまいません。

カードの順位は、2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, Aの順に強くなっていきます（ただし違うスートのものはもっとも弱い）。1巡した時点でもっとも強いカードを出した人が次の親になります。親は取った場札を自分のところに寄せます（手札には加えない）。このなかにハートのカードがあったら、1枚につき1点のペナルティとなります。

手札がなくなった時点でペナルティの計算をし、もっとも少ない人が次のゲームの



最初の親になります。

1枚もハートを取らないことをクリアとします。クリアのときはほかの人のペナルティ分13点がもらえます。クリアが2人のときは、6点ずつで余った1点は次回に持ち越されます。

要するにハートのカードを取らないようにすればいいわけですが、ひとりでもハートを13枚集めた場合だけは例外で、ほかの人から4点ずつもらえます。

だいたい感じがわかったでしょうか？



入力方法

CARD.FNCは1990年5月号で発表されたX68000用カードゲーム支援ツールです。6月号のディスクにも入っていたので、解凍して使ってください。以下にCARD.FNCをBASICに組み込むまでの手順を示します。

まず、ディスクを解凍します。6月号のオマケディスクをBドライブに入れた場合なら、

```
A>LH-E B:GAMES
```

とすると、GAMES.LZHに入ったデータが



Aドライブ上に展開されます。

ここで、BASICからMAKE.BASを実行すると自動的にCARD.FNCというファイルを作成します。できあがったCARD.FNCはBASICが入っているディレクトリに入れてください。

次に、

```
A>ED A:¥BASIC2¥BASIC.CNF
```

のように、エディタでコンフィギュレーションファイルを読み込みます。いちばん下の行に、

```
FUNC=CARD
```

と書き加え、ESC・Eでセーブします。これでCARD.FNCが組み込まれました。

次にBASICを起動してゲーム本体を打ち込みます。全部打ち込んだらRUNでゲームを始めてください。

* * *

プログラムのシャッフル部分はCARD.FNCのサンプルである“99”からルーチンを拝借しました。このルーチンは私が使っていたものよりもよく混ざるようです。

今後もCARD.FNCを使ったトランプゲームを作っていきたいと思いますので、皆さんよろしくお願ひします。

リスト1

```
10 /*
20 /* HEART
30 /* programed by M.I.,May22'90
40 /*
50 screen 1,1,1:console,,0
60 int jj,b1,b2,b3,b4,bb,m,f=0,rd=1
70 dim int cc(51),c(3,12),pp(51),p(3,12),gg(3)
80 dim int h(12),b(3),mai(3),kei(3),kuri(4),ten(4,3)
90 dim str nam(3)={"太郎","次郎","花子","貴方"}
100 palet(1,0)
110 /* main program
120 while f<>1
```

```
130 scrn()
140 play()
150 jd3()
160 endwhile
170 owari()
180 end
190 /* screen
200 func scrn()
210 int i
220 apage(3):vpage(15)
230 fill(0,0,511,511,3)
240 apage(2)
```



```

250 box(0,0,511,511,15):box(1,1,510,510,15)
260 line(2,144,509,144,15,&HFFFF)
270 line(160,2,160,143,15,&HFFFF)
280 line(320,2,320,143,15,&HFFFF)
290 line(2,384,509,384,15,&HFFFF)
300 line(280,145,280,383,15,&HFFFF)
310 for i=0 to 4
320   line(321,i*24+24,509,i*24+24,15,&HFFFF)
330 next
340 for i=0 to 5
350   line(i*30+360,25,i*30+360,143,15,&HFFFF)
360 next
370 symbol(26,42,"H E A R T",1,4,1,1,0)
380 symbol(24,40,"H E A R T",1,4,1,1,0)
390 symbol(376,6,"SCORE",2,1,1,15,0)
400 for i=0 to 3: symbol(324,i*24+53,nam(i),1,1,1,15,0):next
410 for i=1 to 4: symbol(i*30+341,29,strings(i),1,1,1,15,0):next
420 symbol(481,29,"合計",1,1,1,15,0)
430 symbol(348,154,"各自のハート",1,1,1,15,0)
440 for i=0 to 3
450   symbol((i mod 2)*112+286,(i * 2)*104+204,nam(i),1,1,1,
15,0)
460 next
470 symbol(8,428,nam(3),1,1,1,15,0)
480 for i=0 to 3: key(i)=0:next
490 for i=0 to 4: kuri(i)=0:next
500 endfunc
510 /* play
520 func play()
530   while rd<5
540     prep()
550     splay()
560     jd2()
570   endwhile
580 endfunc
590 /*
600 func prep()
610   int i,j,a,b,k,s
620 /* music data set
630   if rd=1 then {
640     m_init()
650     for i=1 to 8: m_alloc(i,2000):m_assign(i,i):next
660     m_trk(1,"q8023v1003t180132 e")
670     m_trk(2,"q8023v1003t180132 c")
680     m_trk(3,"q8032v1002t10014 a")
690     m_trk(4,"q5056v1005t100116aene")
700     m_trk(5,"q7056v1005t 8014 a")
710     m_trk(6,"q8056v1006t 5012 c")
720     m_trk(7,"q601 v1004t1800c#8112dec#dec#dec#d8e8fgefgefge
f8g8ab-gab-gab-ga4")
730     m_trk(8,"q7019v1004t 55b8.b16<d4>a8.b16<c>b8b8116agf#g
ad4b8.b16<d4>a8.b16<c4>b8b8116ab<c>ag4")
740   }
750 /* deal
760   apage(1)
770   fill((rd-1)*30+331,25,(rd-1)*30+359,47,0)
780   fill(rd*30+331,25,rd*30+359,47,5)
790   for i=1 to 4: symbol(i*30+341,29,strings(i),1,1,1,15,0):next
800   if rd=1 then {
810     symbol(184,40,"ルールの説明は",1,1,1,15,0)
820     s=sel(176,96,1,1):if s=1 then rule()
830   }
840   randomize(val(mids(times$4,2)+right$(times$,2)))
850   for i=0 to 51:cc(i)=i:next
860   for i=0 to 3:mai(i)=0:next:m=13
870   if s=2 or rd>1 then {
880     er_upms()
890     symbol(200,24,"シャッフル",1,1,1,15,0)
900     symbol(224,56,"及び",1,1,1,15,0)
910     symbol(200,88,"カード配布",1,1,1,15,0)
920   }
930   fill(40,168,240,200,5)
940   symbol(64,176,"ちょっと待って下さい",1,1,1,15,0)
950   for i=0 to 12:h(i)=0:next
960   for i=0 to 99
970     a=int(rnd()*52):b=int(rnd()*52)
980     k=cc(a):cc(a)=cc(b):cc(b)=k
990   next
1000 fill(40,168,68,200,3)
1010 for i=0 to 51
1020   if cc(i)=1 then pp(i)=13:continue
1030   if cc(i)=14 then pp(i)=26:continue
1040   if cc(i)=27 then pp(i)=39:continue
1050   if cc(i)=40 then pp(i)=52:continue
1060   pp(i)=cc(i)-1
1070 next
1080 fill(69,168,96,200,3)
1090 for i=0 to 12
1100   c(0,i)=cc(i) : p(0,i)=pp(i)
1110   c(1,i)=cc(i+13):p(1,i)=pp(i+13)
1120   c(2,i)=cc(i+26):p(2,i)=pp(i+26)
1130   c(3,i)=cc(i+39):p(3,i)=pp(i+39)
1140 next
1150 for i=0 to 11
1160   for j=i+1 to 12
1170     for k=0 to 3
1180       if p(k,i)>p(k,j) then {
1190         a=p(k,i):p(k,i)=p(k,j):p(k,j)=a
1200         a=c(k,i):c(k,i)=c(k,j):c(k,j)=a
1210       }
1220     next
1230   next
1240   fill(96,168,96+(i+1)*12,200,3)
1250 next
1260 er_upms()
1270 plcd()
1280 if s=1 then {
1290   click()
1300   apage(0):fill(0,0,511,511,0):apage(1)
1310 }
1320 mkba():htmai()
1330 /* play order
1340 if rd=1 then {
1350   symbol(184,24,"順番を決めます",1,1,1,15,0)
1360   symbol(176,56,"いい時にマウスを",1,1,1,15,0)
1370   symbol(200,88,"クリック",1,1,1,15,0)

```

```

1380 mouse(1)
1390 symbol(140,177,"が最初",1,1,1,1,0)
1400 repeat
1410   jj=int(rnd()*4)
1420   fill(96,177,128,192,15)
1430   symbol(96,177,nam(jj),1,1,1,1,0)
1440   msstat(i,j,a,b)
1450   until a<>0 or b<>0
1460   mouse(0)
1470   wait(50):er_upms()
1480 } else symbol(96,177,nam(jj)+"が最初",1,1,1,1,0):wait(60)
1490 endfunc
1500 /* play
1510 func splay()
1520   repeat
1530     bl=0:b2=0:b3=0:b4=0:bb=1
1540     asplay()
1550     jdl()
1560     m=m-1
1570   until m=0
1580 endfunc
1590 /*
1600 func ssplay()
1610   while bb<5
1620     if jj=0 and jj<=2 then {
1630       if bb=1 then {
1640         com1():bb=2:jj=jj+1
1650         if jj=3 then you():jj=0:bb=3:continue
1660       } else if bb=2 then {
1670         com(2,b2):bb=3:jj=jj+1
1680         if jj=3 then you():jj=0:bb=4:continue
1690       } else if bb=3 then {
1700         com(3,b3):bb=4:jj=jj+1
1710         if jj=3 then you():bb=5
1720       } else if bb=4 then com(4,b4):bb=5
1730     } else if jj=3 and bb=1 then you():jj=0:bb=2:continue
1740   endwhile
1750 endfunc
1760 /* com play as 1st player
1770 func com1()
1780   int i,ss,hm=0,sm=0,bc=0
1790   dsban(jj)
1800   for i=0 to m-1
1810     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
1820     if p(jj,i)<14 then sm=sm+1
1830   next
1840   while bc=0
1850     if hm>0 then {
1860       i=sm
1870       if p(jj,i)<17 then bc=1:break
1880       if p(jj,i)<18 and h(0)+h(1)+h(2)>=1 then bc=1:break
1890       if p(jj,i)<19 and h(0)+h(1)+h(2)>=2 then bc=1:break
1900       if p(jj,i)<20 and h(0)*h(1)*h(2)>0 then bc=1:break
1910       if p(jj,i)<21 and h(0)*h(1)*h(2)*h(3)>0 then bc=1:br
1920     }
1930     if p(jj,i)<22 and h(0)*h(1)*h(2)*h(3)*h(4)>0 then bc
=0:break
1940   }
1950   repeat
1960     i=int(rnd()*m)
1970     until p(jj,i)<14 or p(jj,i)>26
1980     bc=1
1990   endwhile
2000   is=i:bacd(jj,is)
2010   bl=p(jj,is):p(jj,is)=0:c(jj,is)=0:b(jj)=bl
2020   if bl>13 and bl<27 then gg(jj)=bl:h(bl-14)=1
2030   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2040   cdleft(is)
2050   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2060   if jj=2 then wait(15) else wait(30)
2070 endfunc
2080 /* com play as 2nd to 4th player
2090 func com(q,id)
2100   int i,ss,hm=0,sm=0,ap=0,bc=0
2110   dsban(jj)
2120   for i=0 to m-1
2130     if p(jj,i)>13 and p(jj,i)<27 then hm=hm+1
2140     if p(jj,i)<14 then sm=sm+1
2150     if (p(jj,i)-1)*13=(bl-1)*13 then ap=ap+1
2160   next
2170   while bc=0
2180     if ap=0 then {
2190       if sm+hm>0 then is=sm+hm-1:b(jj)=0:bc=1:break else {
2200         is=m-1:b(jj)=0:bc=1:break }
2210     }
2220     if ap>0 and (bl>13 and bl<27) then {
2230       switch q
2240         case 2:is=scom2(hm,sm):break
2250         case 3:is=scom3(hm,sm):break
2260         case 4:is=scom4(hm,sm)
2270       endswitch
2280       if is>0 then bc=1
2290       if bc=0 and p(jj,sm)<22 then is=sm:bc=1:break
2300       if bc=0 and p(jj,sm)>21 then is=sm+hm-1:bc=1:break
2310     }
2320     if ap>0 and (bl<14 or bl>26) then {
2330       switch q
2340         case 2:is=sscom2():break
2350         case 3:is=sscom3():break
2360         case 4:is=sscom4()
2370       endswitch
2380       bc=1
2390     }
2400   endwhile
2410   bacd(jj,is)
2420   id=p(jj,is):p(jj,is)=0:c(jj,is)=0
2430   if ap>0 then b(jj)=id
2440   if id>13 and id<27 then gg(jj)=id:h(id-14)=1
2450   for i=0 to m-1:cc(i)=c(jj,i):pp(i)=p(jj,i):next
2460   cdleft(is)
2470   for i=0 to m-1:c(jj,i)=cc(i):p(jj,i)=pp(i):next
2480   if jj=2 then wait(15) else wait(30)
2490 endfunc
2500 /*play you
2510 func you()

```



```

2520 int i,is,x,y,l,r,ap=0,bc=0
2530 dsban(3)
2540 if b1>0 then {
2550   for i=0 to m-1
2560     if (p(3,i)-1)*13 = (b1-1)*13 then ap=ap+1
2570     next
2580 }
2590 while bc=0
2600   symbol(176,48,"出したいカードを",1,1,1,15,0)
2610   symbol(208,84,"クリック",1,1,1,15,0)
2620   mouse(1)
2630   msarea(49,401,502,495):setmpos(64,432)
2640   repeat
2650     mstat(x,y,l,r)
2660     until l<>0 or r<>0
2670     mpos(x,y)
2680     mouse(0):er_upms()
2690     if m>9 then is=(x-48)*34 else is=(x-48)*50
2700     if is>m-1 then dame():wait(40):er_upms():continue
2710     if b1>0 and ap>0 and (p(3,is)-1)*13 <> (b1-1)*13 then {
2720       dame():wait(40):er_upms():continue
2730     } else bacd(3,is):b(3)=p(3,is):bc=1
2740     if b1>0 and ap=0 then b(3)=0
2750     switch bb
2760       case 1: b1=p(3,is):break
2770       case 2: b2=p(3,is):break
2780       case 3: b3=p(3,is):break
2790       case 4: b4=p(3,is)
2800     endswitch
2810     if p(3,is)>13 and p(3,is)<27 then gg(3)=p(3,is):h(gg(3)-14)=1
2820     p(3,is)=0:c(3,is)=0
2830     for i=0 to m-1:cc(i)=c(3,i):pp(i)=p(3,i):next
2840     cdleft(is)
2850     for i=0 to m-1:c(3,i)=cc(i):p(3,i)=pp(i):next
2860     fill(2,385,509,509,0):m=m-1
2870     plcd(i):m=m+1
2880   endwhile
2890 endfunc
2900 /* judge1
2910 func jd1()
2920   int i,j,a
2930   dim int ba(3)
2940   for i=0 to 3:ba(i)=b(i):next
2950   for i=0 to 2
2960     for j=i+1 to 3
2970       if b(i)<b(j) then a=b(i):b(i)=b(j):b(j)=a
2980     next
2990   next
3000   for jj=0 to 3
3010     if b(0)=ba(jj) then kachi(jj):wait(40):break
3020   next
3030   a=mai(jj)
3040   for i=0 to 3
3050     if gg(i)>0 then {
3060       mai(jj)=mai(jj)+1:htcd(jj,i):htmai(i):gg(i)=0
3070     next
3080     if mai(jj)>a then ha(mai(jj)-a):wait(40) else ha(0):wait
3090   }
3100   mkba()
3110 endfunc
3120 /* judge2
3130 func jd2()
3140   int i,j,a,b,cla
3150   m_play(6)
3160   apage(0)
3170   fill(2,145,509,383,8)
3180   box(64,208,448,352,15,&HFFFF)
3190   line(65,256,447,256,15,&HFFFF)
3200   for i=0 to 2:line(65,i*24+280,239,i*24+280,15,&HFFFF)
3210   line(313,i*24+280,375,i*24+280,15,&HFFFF)
3220   next
3230   line(120,209,120,352,15,&HFFFF)
3240   line(176,209,176,352,15,&HFFFF)
3250   line(240,209,240,352,15,&HFFFF)
3260   line(312,209,312,352,15,&HFFFF)
3270   line(376,209,376,352,15,&HFFFF)
3280   symbol(160,168,"第"+str$(rd)+"回 得点計算",1,1,2,15,0)
3290   symbol(124,217,"ハート",1,1,1,15,0)
3300   symbol(124,237,"枚 数",1,1,1,15,0)
3310   symbol(184,217,"今 回 の",1,1,1,15,0)
3320   symbol(184,237,"得 点",1,1,1,15,0)
3330   symbol(244,217,"前回から",1,1,1,15,0)
3340   symbol(244,237,"の繰越点",1,1,1,15,0)
3350   symbol(320,217,"修正後",1,1,1,15,0)
3360   symbol(320,237,"の得点",1,1,1,15,0)
3370   symbol(380,217,"次回への",1,1,1,15,0)
3380   symbol(388,237,"繰越点",1,1,1,15,0)
3390   for i=0 to 3
3400     symbol(76,i*24+261,nam(i),1,1,1,15,0)
3410     if mai(i)>9 then {
3420       symbol(132,i*24+261,str$(mai(i)),2,1,1,15,0)
3430     } else symbol(148,i*24+261,str$(mai(i)),2,1,1,15,0)
3440     if mai(i)=0 then cla=cla+1
3450   next
3460   if cla=0 then {
3470     kuri(rd)=13+kuri(rd-1)
3480     for i=0 to 3:ten(rd,i)=-mai(i):next
3490   }
3500   if cla=3 then {
3510     kuri(rd)=0
3520     for i=0 to 3
3530       if mai(i)=13 then ten(rd,i)=12 else ten(rd,i)=-4
3540     next
3550   }
3560   if cla>0 and cla<3 then {
3570     kuri(rd)=13 mod cla + kuri(rd-1) mod cla
3580     for i=0 to 3
3590       if mai(i)=0 then ten(rd,i)=13*cla else ten(rd,i)=-ma
3600   }
3610   next
3620   symbol(264,297,str$(kuri(rd-1)),2,1,1,15,0)
3630   symbol(398,297,str$(kuri(rd)),2,1,1,15,0)
3640   for i=0 to 3
3650     if ten(rd,i)>0 then {
3660       symbol(200,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3670     } else symbol(184,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3680   next
3690   if cla=3 then {
3700     for i=0 to 3
3710       if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1):b
3720     next
3730   }
3740   if cla>0 and cla<3 then {
3750     for i=0 to 3
3760       if ten(rd,i)>0 then ten(rd,i)=ten(rd,i)+kuri(rd-1):vc
3770     next
3780   }
3790   for i=0 to 3
3800     kei(i)=kei(i)+ten(rd,i)
3810     if ten(rd,i)>0 then {
3820       symbol(336,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3830       symbol(rd*30+342,i*24+53,str$(ten(rd,i)),1,1,1,15,0)
3840     } else {
3850       symbol(320,i*24+261,str$(ten(rd,i)),2,1,1,15,0)
3860       symbol(rd*30+334,i*24+53,str$(ten(rd,i)),1,1,1,15,0)
3870     }
3880   fill(481,i*24+49,509,i*24+71,0)
3890   if kei(i)>0 then {
3900     symbol(493,i*24+53,str$(kei(i)),1,1,1,15,0)
3910   } else symbol(485,i*24+53,str$(kei(i)),1,1,1,15,0)
3920 next
3930 rd=rd+1
3940 if rd<5 then {
3950   ten(rd-1,3)=ten(rd-1,3)+1
3960   for i=0 to 3:cc(i)=ten(rd-1,i):next
3970   for i=0 to 2
3980     for j=i+1 to 3
3990       if cc(i)<cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
4000   next
4010   for i=0 to 3
4020     if cc(0)=ten(rd-1,i) then jj=i:break
4030   next
4040   ten(rd-1,3)=ten(rd-1,3)-1
4050   symbol(176,40,str$(rd)+" 回目を始めます",1,1,1,15,0)
4060   s=sel(176,96,2,2)
4070   if s=2 then f1:rd=5
4080 } else click()
4090 apage(1):fill(0,0,511,511,0)
4100 apage(0):fill(0,144,511,511,0):er_upms():apage(1)
4110 endfunc
4120 /* judge3
4130 func jd3()
4140   int i,j,a,b
4150   if f<1 then {
4160     vpage(9)
4170     apage(0):fill(0,0,511,511,0)
4180     for i=0 to 5
4190       box(48+i*6,80+i*6,464-i*6,432-i*6,15)
4200     next
4210     fill(79,111,433,401,2)
4220     kei(3)=kei(3)+1
4230     for i=0 to 3:cc(i)=kei(i):next
4240     for i=0 to 2
4250       for j=i+1 to 3
4260         if cc(i)<cc(j) then a=cc(i):cc(i)=cc(j):cc(j)=a
4270       next
4280     next
4290     for i=0 to 3
4300       if cc(0)=kei(i) then jj=i:break
4310     next
4320     symbol(97,218,nam(jj)+" の優勝!",2,2,2,5,0):m_play(7)
4330     symbol(352,440,"もう 1 度やりませんか",1,1,1,15,0)
4340     s=sel(380,465,2,2)
4350     if s=1 then {
4360       rd=1:fill(0,0,511,511,0)
4370       apage(1):fill(0,0,511,511,0)
4380       apage(2):fill(0,0,511,511,0)
4390       vpage(15)
4400     } else f=1
4410   }
4420 endfunc
4430 /* owari
4440 func owari()
4450   vpage(2):apage(1)
4460   fill(0,0,511,511,2)
4470   symbol(272,400,"お疲れ様でした",1,1,2,15,0)
4480   m_play(8)
4490 endfunc
4500 /*
4510 func scom2(hm,sm)
4520   int i,is,bc=0
4530   for i=0 to hm-1
4540     is=sm+hm-1-i
4550     if p(jj,is)<b1 then bc=1:break
4560   next
4570   if bc=0 then is=0
4580   return(is)
4590 endfunc
4600 /*
4610 func scom3(hm,sm)
4620   int i,is,bc=0
4630   for i=0 to hm-1
4640     is=sm+hm-1-i
4650     if p(jj,is)<b1 or p(jj,is)<b2 then bc=1:break
4660   next
4670   if bc=0 then is=0
4680   return(is)
4690 endfunc
4700 /*
4710 func scom4(hm,sm)
4720   int i,is,bc=0
4730   for i=0 to hm-1
4740     is=sm+hm-1-i
4750     if p(jj,is)<b1 or p(jj,is)<b2 or p(jj,is)<b3 then bc=1
4760   :break
4770   next
4780   if bc=0 then is=0

```



```

4780 return(is)
4790 endfunc
4800 /*
4810 func sscom2()
4820 int i, is
4830 for i=0 to m-1
4840 is=m-1-i
4850 if (p(jj, is)-1)*13=(b1-1)*13 then break
4860 next
4870 return(is)
4880 endfunc
4890 /*
4900 func sscom3()
4910 int i, is, a=0
4920 if b2>13 and b2<27 then {
4930 for i=0 to m-1
4940 is=m-1-i
4950 if (p(jj, is)-1)*13=(b1-1)*13 and p(jj, is)<b1 then a=
1:break
4960 next
4970 if a=0 then {
4980 for i=0 to m-1
4990 is=m-1-i
5000 if (p(jj, is)-1)*13=(b1-1)*13 then break
5010 next
5020 }
5030 } else is=sscom2()
5040 return(is)
5050 endfunc
5060 /*
5070 func sscom4()
5080 int i, is, a=0
5090 if (b2>13 and b2<27) and (b3>13 and b3<27) then {
5100 for i=0 to m-1
5110 is=m-1-i
5120 if (p(jj, is)-1)*13=(b1-1)*13 and p(jj, is)<b1 then a=
1:break
5130 next
5140 if a=0 then {
5150 for i=0 to m-1
5160 is=m-1-i
5170 if (p(jj, is)-1)*13=(b1-1)*13 then break
5180 next
5190 }
5200 } else if (b2>13 and b2<27) and (b3>14 or b3>26) then {
5210 for i=0 to m-1
5220 is=m-1-i
5230 if (p(jj, is)-1)*13=(b1-1)*13 and (p(jj, is)<b1 or p(j
j, is)<b3) then a=1:break
5240 next
5250 if a=0 then {
5260 for i=0 to m-1
5270 is=m-1-i
5280 if (p(jj, is)-1)*13=(b1-1)*13 then break
5290 next
5300 }
5310 } else if (b2>14 or b2>26) and (b3>13 and b3<27) then {
5320 for i=0 to m-1
5330 is=m-1-i
5340 if (p(jj, is)-1)*13=(b1-1)*13 and (p(jj, is)<b1 or p(j
j, is)<b2) then a=1:break
5350 next
5360 if a=0 then {
5370 for i=0 to m-1
5380 is=m-1-i
5390 if (p(jj, is)-1)*13=(b1-1)*13 then break
5400 next
5410 }
5420 } else is=sscom2()
5430 return(is)
5440 endfunc
5450 /*
5460 func cdleft(k)
5470 int i
5480 for i=0 to m-k:cc(k+i)=cc(k+i+1):pp(k+i)=pp(k+i+1):next
5490 endfunc
5500 /*
5510 func sel(x, y, m, n)
5520 int i, j, a, b
5530 str mm, nn
5540 switch m
5550 case 1:mm=" 必 要":break
5560 case 2:mm=" オ K"
5570 endswitch
5580 switch n
5590 case 1:nn=" 不 要":break
5600 case 2:nn=" やめる"
5610 endswitch
5620 fill(x, y, x+56, y+24, 15):fill(x+72, y, x+128, y+24, 15)
5630 symbol(x+4, y+4, mm, 1, 1, 1, 1, 0)
5640 symbol(x+76, y+4, nn, 1, 1, 1, 1, 0)
5650 mouse(1)
5660 msarea(x+1, y+1, x+127, y+23)
5670 setmspos(x+28, y+8)
5680 repeat
5690 msstat(i, j, a, b)
5700 until a<>0 or b<>0
5710 mspos(i, j)
5720 mouse(0)
5730 if i<x+64 then {
5740 fill(x, y, x+56, y+24, 1):symbol(x+4, y+4, mm, 1, 1, 1, 1, 15, 0):s=1
5750 symbol(x+4, y+4, mm, 1, 1, 1, 1, 15, 0):a=1
5760 } else {
5770 fill(x+72, y, x+128, y+24, 1)
5780 symbol(x+76, y+4, nn, 1, 1, 1, 1, 15, 0):s=2
5790 }
5800 return(s):wait(40)
5810 endfunc
5820 /*
5830 func click()
5840 int i, j, a, b
5850 symbol(176, 48, "よければマウスを", 1, 1, 1, 15, 0)
5860 symbol(208, 84, "クリック", 1, 1, 1, 15, 0)
5870 mouse(1)

```

```

5880 msarea(176, 48, 288, 96)
5890 setmspos(232, 70)
5900 repeat
5910 msstat(i, j, a, b)
5920 until a<>0 or b<>0
5930 mouse(0)
5940 er_upms()
5950 endfunc
5960 /*
5970 func mkba()
5980 fill(3, 145, 279, 383, 8):fill(40, 168, 240, 200, 15)
5990 for i=0 to 3:symbol(i*56+40, 344, nam(i), 1, 1, 1, 15, 0):next
6000 endfunc
6010 /*
6020 func dsban(j)
6030 er_ms():symbol(108, 177, nam(j)+ "の番", 1, 1, 1, 1, 0)
6040 endfunc
6050 /*
6060 func kachi(j)
6070 er_ms():symbol(48, 177, nam(j)+ "の勝ち", 1, 1, 1, 5, 0)
6080 endfunc
6090 /*
6100 func ha(ht)
6110 symbol(134, 177, "ハート "+str$(ht)+"枚", 1, 1, 1, 1, 0):m_play
(4)
6120 endfunc
6130 /*
6140 func ha0()
6150 symbol(134, 177, "ハート無し", 1, 1, 1, 1, 0):m_play(5)
6160 endfunc
6170 /*
6180 func dame()
6190 er_upms()
6200 symbol(200, 48, "出せません", 1, 1, 1, 5, 0):m_play(3)
6210 endfunc
6220 /*
6230 func baed(j, i)
6240 c_put(j*56+32, 225, c(j, i)):m_play(1, 2)
6250 endfunc
6260 /*
6270 func plcd()
6280 int i
6290 if m>9 then {
6300 for i=0 to m-1
6310 c_put(i*34+48, 400, c(3, i))
6320 line(i*34+47, 400, i*34+47, 496, 1)
6330 m_play(1, 2)
6340 next
6350 } else {
6360 for i=0 to m-1
6370 c_put(i*50+48, 400, c(3, i))
6380 m_play(1, 2)
6390 next
6400 }
6410 symbol(16, 453, str$(m), 1, 1, 1, 15, 0)
6420 endfunc
6430 /*
6440 func htcd(jj, s)
6450 int a, b, h
6460 a=(jj mod 2)*112+317:b=(jj % 2)*104+176
6470 if gg(s)=26 then h=14 else h=gg(s)+1
6480 c_put(a+mai(jj)*2, b, h)
6490 line(a+mai(jj)*2-1, b, a+mai(jj)*2-1, b+96, 1)
6500 m_play(1, 2)
6510 endfunc
6520 /*
6530 func htmai()
6540 int i
6550 for i=0 to 3
6560 fill((i mod 2)*112+294, (i % 2)*104+228, (i mod 2)*112+3
10, (i % 2)*104+244, 0)
6570 symbol((i mod 2)*112+294, (i % 2)*104+228, str$(mai(i)),
1, 1, 1, 15, 0)
6580 next
6590 endfunc
6600 /*
6610 func wait(t)
6620 int i
6630 for i=0 to t*100:next
6640 endfunc
6650 /*
6660 func er_upms()
6670 fill(161, 3, 319, 143, 0)
6680 endfunc
6690 /*
6700 func er_ms()
6710 fill(40, 168, 240, 200, 15)
6720 endfunc
6730 /*
6740 func rule()
6750 apage(0)
6760 fill(2, 145, 509, 383, 8)
6770 symbol(196, 160, "ル - ル", 1, 1, 1, 15, 0)
6780 symbol(60, 190, "1 : カードの強さは A, K, Q, J...4, 3, 2 の順",
1, 1, 1, 15, 0)
6790 symbol(60, 208, "2 : 各自が 1 枚ずつ同じ種類を出さねばなり
ません", 1, 1, 1, 15, 0)
6800 symbol(120, 225, "3 : 但し手持ちが無ければ、何でもかまいません",
1, 1, 1, 15, 0)
6810 symbol(60, 243, "4 : 最も強い札を出した人が 4 枚全部取りま
す", 1, 1, 1, 15, 0)
6820 symbol(60, 261, "5 : 取った中にハートがあれば 1 枚につき -
1 点", 1, 1, 1, 15, 0)
6830 symbol(60, 279, "6 : 持ち札が無くなるまで繰り返します", 1, 1,
1, 15, 0)
6840 symbol(60, 297, "7 : プラス点はハートを取らなかった人で分
けず", 1, 1, 1, 15, 0)
6850 symbol(120, 314, "1 人の時... + 13 点", 1, 1, 1, 15, 0)
6860 symbol(120, 331, "2 人の時... + 6 点、残りは繰り越し",
1, 1, 1, 15, 0)
6870 symbol(60, 349, "7 : 1 人でハート 13 枚取れば +12 点、他の
人は -4 点", 1, 1, 1, 15, 0)
6880 apage(1)
6890 endfunc

```




トランジェントコマンドを作る

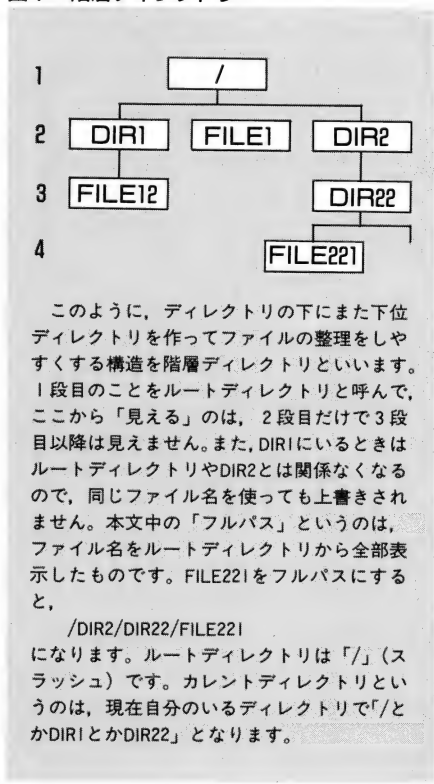
亀田 雅彦 Kameda Masahiko

KAME-DOSをもっとDOSらしく使うための方法として、KAME-DOSの外部コマンドを作成してみましょう。ディスク管理のほか、さまざまなプログラムがコマンドとして使用できます。こういったものがBASICで記述できるのです。

先月号でノーマルX1にも対応して、いよいよ本格的になってきました。もし、X1ユーザーでまだKAME-DOSを入手していない方は、ぜひバックナンバーなどから入力するようにしてください。

6、7月号のプログラムだけではなかなかその威力を発揮しないKAME-DOSも、今月から紹介していく外部コマンドを活用すればその世界が広がります。特にノーマルX1ユーザーには、ディスク関係の命令がturboBASICに匹敵するようになるので、お楽しみに。また、外部コマンドのノウハウが蓄積していくとユーザー自身の手でKAME-DOSワールドを広げていくことができるようになります（もちろん最初の公約どおり、BASICで）。

図1 階層ディレクトリ



とりあえず、今月から何回かに分けて、普通のDOSにあるような命令を外部コマンドとして発表しながら、その動作と作り方を説明しましょう。基本的に外部コマンドも内部コマンドも(COMMAND.X1内に用意されてるもの)、作りは同じなのでCOMMAND.X1の理解の助けにもなると思います。

それでは、今月は「MD.X1」「RD.X1」そして、特集と関連して「GLOAD.X1」「GSAVE.X1」を発表してみましょう。

外部コマンドワールド

リスト3が「MD.X1」です。turboBASICというところのMKDIRにあたります。下位ディレクトリをカレントディレクトリの下に作るのですが、難しいところなので階層ディレクトリ全般について簡単に図1で説明しておきます。また、階層ディレクトリとは切っても切れない関係にあるCD（ディレクトリの変更）命令については、内部コマンドなので6月号に解説されています（でも、6月号ではちょっと手抜き解説が多かったと反省することしきりです）。

反省ばかりしていても進歩がないので、さっそく使い方に入りましょう。

命令：MD (MKDIR)

書式：MD 新規ディレクトリネーム

プログラム：リスト3

まずリスト3を打ち込んでください。使用BASICは、いま自分の持っているINTEGRAL.Xが動いているものならなんでも大丈夫です（CZ-8FB01ver1.0, turboBASIC, Z-BASIC）。用途別に自分でBASICシステムを構築してください。ただしCZ-8FB01では日本語入力ができないので、リストの一番最後にDATA文としてまとめら

れてるメッセージは、注釈行のほうを生かして日本語のほうは打ち込まなくて結構です。たとえばリストで、

1650 LABEL "d1": DATA エラーが発生しました!!

1660 'LABEL "d1": DATA Error !!
という2行は、

1650 '

1660 LABEL "d1": DATA Error !!

というようにします。これが2行ずつ組になっているので、それぞれについて変更してください。日本語入力ができて、しかも使用中に日本語表示ができる（をしたい）場合には（ディスプレイの関係で表示できないこともある）、そのまま入力してください。以後、外部コマンドの入力形式はだいたい同じようなかたちになります。

使い方：

入力したら、カレントドライブかパスの通っているドライブにセーブしてください。INTEGRAL Xのコマンドライン（[X:/]の状態）から、セーブしたときのファイル名（この場合は「MD.X1」か「MKDIR.X1」）をタイプしてリターンキーを押してください。「MD」か「MKDIR」だけで、拡張子はいりません。

6月号でも書いたことですが、拡張子が「.X1」のBASICファイルはKAME-DOSの外部コマンドとして認識されます。見かけ上は、内部コマンドの実行となんら変わりありません。また、コマンドラインからパラメータとして与えられる新規ディレクトリネームの書式については、囲みを参照してください。

パスが通ってなかったり、ファイル名をタイプミスしたときはエラーになります。エラーが起きずに、しばらくすると外部コマンドがロードされて起動します。指定に

間違いがなければ、下位ディレクトリを作成して、パスに従って「COMMAND.X1」をロードしなおしてコマンドラインに復帰します（CP/Mでいうリブート）。ここで、外部コマンド実行の際の注意点を挙げておきましょう。

1) 外部コマンドのファイル名は、内部コマンドのコマンド名にあたるものなのでわかりやすくすること（片仮名などにするとあとで苦勞します）。拡張子は「.X1」にすること。

2) 外部コマンド実行中にブレイクして実行を強制的に中止したときは、必ず「COMMAND.X1」を実行するところから始めてください。外部コマンドをブレイクしてそのままRUNすると、変数がクリアされるので最悪の場合暴走します。これは入力したプログラムをデバッグしているときも同じことで、エラーが出て止まったら、入力ミスを訂正していったんセーブして、「COMMAND.X1」をRUNしてそのコマンドラインから外部コマンドを実行するようにしてください（図2）。

3) 「COMMAND.X1」は必ずパスの通っているドライブにセーブしておいてください。リブートするときにパスの順に従って「COMMAND.X1」を探すので、みつからないと「リブートできません」というメッセージが出て実行が止まります。コマンドラインからの実行のときとは違って、カレントドライブでもパスが通ってないと探しにいきません。

4) 外部コマンドからリブートした時点で、下位ディレクトリにいてもすべてルートディレクトリに戻されます。たとえば、[A:/TEST/] から「MD」を実行して戻ってくると、[A:/] になっているということ

です。これが外部コマンドと内部コマンドが見かけ上異なる唯一の点です。

上記のうち、特に2)が大切なので必ず守ってください。このほかにも外部コマンド実行中にさまざまなエラーが発生する可能性があります。その場合はエラーメッセージを出力し、実行を中止して「COMMAND.X1」へ復帰しようとします。エラーメッセージは個々の外部コマンドが持っているものなので、統一されていません。

以上のことは、外部コマンド全般についていえることなので、これからも覚えておいてください。

命令：RD (RMDIR)

書式：RD 消去するディレクトリネーム

プログラム：リスト4

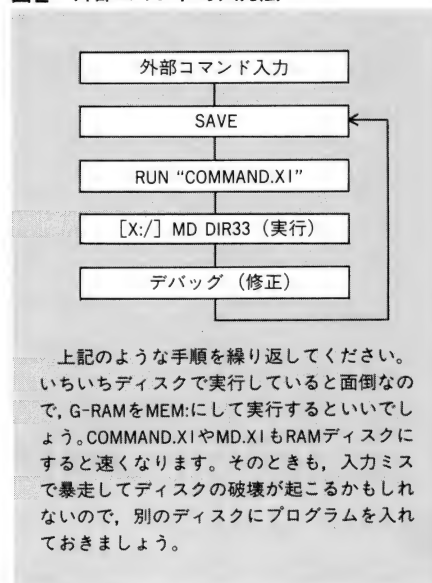
MDの逆で既存のディレクトリを消します。使用BASICも、その日本語部分の入力の仕方もMDと同じです。

使い方：

セーブする際の注意や、実行の仕方についてもMDのところを見てください。機能的にはturboBASICのものと同じです。ディレクトリ内にファイルが残っているときは、消去できません。ファイルをDELコマンドですべて消してから実行してください。

以上、2つの外部コマンドを紹介しましたが、これによって作成されたディレクト

図2 外部コマンドの入力法



リなどは完全にBASICとの互換性があるので、ファイルのやりとりも自由にできます。

でも、CZ-8FB01には階層ディレクトリ機能がないので、KAME-DOSで作った下位ディレクトリにはBASIC側からはアクセスできません。なお、BASICのみならずMS-DOSフォーマット（2D、2HD）とも互換性があるので、MS-DOSディスクにディレクトリを作成しようとすれば自動的にプログラム側で判断して、MS-DOSフォーマ

ファイルネーム

ディレクトリの名前も、基本的にそのディスクフォーマットのファイル名と同じです。ファイル名の方は各マニュアルをみてもらうなり、6月号にも少し解説しておきました。turboBASICの場合ディレクトリの拡張子は「.DIR」になるので、それにあわせておきました。フルパスで指定もできますし、カレントドライブからの指定もできます。図1のDIR22の下にDIR33を作りたいのなら、ルートから「MD /DIR2/DIR22/DIR33」か、DIR22から「MD DIR33」です。消したい場合は、MDをRDに変えてください。

リスト1 GLOAD.X1

```
1000 'GLOAD.X1 Ver 1.0          By Kameda
1010 '
1020 DEFINT a-z:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 iomm=PEEK(v_iomm):baddr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1070 MEM$(s_ff,2)=MKI$(&H2000)
1080 bsiz!=&HC0* &H100:MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1090 POKE v_dn,PEEK(s_dn):IF fe$(1)=" THEN "12"
1100 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1110 IF PEEK(&HD07F)=0 THEN dirg=PEEK(&HE139):POKE &HE139,8
1120 '-----({ MAIN ROUTINE )-----
1130 '
1140 GOSUB 1380
1150 k=PEEK(v_stop):IF k=3 THEN "13" ELSE IF k<>0 THEN "1"
1160 CLS:ff$=MEM$(v_fnam+13,3):k=PEEK(&HD07F)
1170 '
1180 IF f$="GL0" OR f$="gl0" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1190 IF f$="GL1" OR f$="gl1" THEN IF k THEN WIDTH 80,25,0,1 ELSE WIDTH 80
1200 IF f$="GM0" OR f$="gm0" THEN IF k THEN WIDTH 40,25,0,2 ELSE WIDTH 40
1210 IF f$="GM1" OR f$="gm1" THEN IF k THEN WIDTH 80,25,0,2 ELSE WIDTH 80
1220 IF f$="GH0" OR f$="gh0" THEN IF k THEN WIDTH 40,25,1,2 ELSE WIDTH 40
1230 IF f$="GH1" OR f$="gh1" THEN IF k THEN WIDTH 80,25,1,2 ELSE WIDTH 80
1240 IF f$="GL2" OR f$="gl2" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1250 IF f$="GL3" OR f$="gl3" THEN IF k THEN WIDTH 40,25,0,1 ELSE WIDTH 40
1260 INIT:IF k THEN POKE v_wfd0,PEEK(&HF8D6)
1270 GOSUB 1380:IF PEEK(v_stop) THEN "1"
1280 '
1290 POKE v_iomm,1:CALL m_devi:IF PEEK(v_stop) THEN "1"
1300 IF PEEK(v_iofg)=0 GOTO 1320
1310 POKE v_iomm,2:CALL m_devi:IF PEEK(v_stop) THEN "1"
1320 '
1330 GOSUB "ending"
1340 POKE &HE137,6:POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=baddr$
1350 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dirg
1360 proces=proces-1:CHAIN proces(proces)
1370 '-----({ OPEN          )-----
1380 '
1390 MEM$(v_badr,2)=MKI$(&H3000)
```



```

1400 POKE v_ddrv+1,7,1:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1410 POKE v_od,1:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1420 IF PEEK(v_stop) RETURN
1430 POKE v_sbdr,1:POKE v_op,0:d$=USR1(fe$)
1440 MEM$(v_badr,2)=MKI$(&H4000):RETURN
1450 '------( END )-----
1460 '
1470 LABEL "ending"
1480 CONSOLE 0,25:CLS:CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1490 REPEAT:A$=INKEY$:UNTIL A$=" "
1500 CLS:RETURN
1510 '------( ERROR ROUTINE )-----
1520 '
1530 LABEL "!4":RESTORE "m3":GOTO 1570
1540 LABEL "!3":RESTORE "m2":GOTO 1570
1550 LABEL "!2":RESTORE "m1":GOTO 1570
1560 LABEL "!" :RESTORE "m0"
1570 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1580 POKE v_stop,0:GOTO 1340
1590 '------( DATA AREA )-----
1600 LABEL "m0":DATA エラーが発生しました!!
1610 'LABEL "m0":DATA Error !!
1620 LABEL "m1":DATA ファイル・ネームを指定してください
1630 'LABEL "m1":DATA Need FILE-NAME
1640 LABEL "m2":DATA ファイルが見つかりません
1650 'LABEL "m2":DATA FILE Not Found
1660 LABEL "m3":DATA リポートできません
1670 'LABEL "m3":DATA Not REBOOT

```

リスト2 GSAVE.X1

```

1000 'GSAVE.X1 Ver 1.0 By Kameda
1010 '
1020 DEFINT a-z:INIT:IF PEEK(&HD07F) THEN KLIST 0
1030 CONSOLE 0,25
1040 DEFUSR1=m_opens:DEFUSR2=m_preop
1050 '
1060 CLS:LOCATE 10,10:PRINT "SAVE GRAM= [1] 96K"
1070 LOCATE 10,12:PRINT " [2] 64K":COLOR 5
1080 LOCATE 10,14:PRINT "[space]=GRAPHIC ON OFF":COLOR 7
1090 LOCATE 10,16:PRINT " PUSH [1] or [2]";
1100 k=0:REPEAT:a$=INKEY$(1):sx=VAL(a$)
1110 IF a$=" " THEN IF k=0 THEN k=1:SCREEN ELSE k=0:PALET
1120 UNTIL 1<=sx AND sx<=2:PRINT sx
1130 '
1140 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1150 MEM$(s_ff,2)=MKI$(&H1800):MEM$(v_badr,2)=MKI$(&H3000)
1160 IF sx=1 THEN bsiz!=&HC0*&H100 ELSE bsiz!=&H60*&H100
1170 MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1180 POKE v_dn,PEEK(s_dn):IF fe$(1)=" " THEN "2"
1190 POKE &HE137,4:POKE v_mac,PEEK(s_mac4+PEEK(v_dn))
1200 IF PEEK(&HD07F)=0 THEN dirg=PEEK(&HE139):POKE &HE139,8
1210 '------( MAIN ROUTINE )-----
1220 '
1230 GOSUB 1520:IF PEEK(v_stop)<>0 THEN ""
1240 MEM$(v_badr,2)=MKI$(&H4000)
1250 '
1260 i=1:k=15:IF m=2 THEN k=1 ELSE IF m=4 THEN k=2
1270 IF sx=2 THEN 1330
1280 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_dev1
1290 IF PEEK(v_stop) THEN ""
1300 POKE v_iomm,2:POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_dev1
1310 IF PEEK(v_stop) THEN ""
1320 fx$=MKI$(&H2000):fm$=MKI$(&H8000)+MKI$(1):GOTO 1390
1330 '
1340 POKE v_iomm,1:POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_dev1
1350 IF PEEK(v_stop) THEN "" ELSE MEM$(v_badr,2)=MKI$(&HA000)
1360 POKE v_od,2:POKE v_iofg,2:POKE v_edr,k:CALL m_dev1
1370 IF PEEK(v_stop) THEN ""
1380 fx$=MKI$(&HC000):fm$=fx$+MKI$(0)
1390 '
1400 z=1:f$=fx$+MKI$(0):IF m=2 OR m=4 THEN z=0:f$=fm$
1410 POKE v_zoku+2,z:MEM$(v_fszi,4)=f$
1420 POKE v_od,2:MEM$(v_badr,2)=MKI$(&H3000):CALL m_saved:CLS
1430 IF PEEK(v_stop) THEN ""
1440 '
1450 CFLASH 1:PRINT "PUSH ANY KEY":CFLASH 0
1460 REPEAT:A$=INKEY$:UNTIL A$<>"":CLS:POKE &HE137,6
1470 IF PEEK(&HD07F)=0 THEN POKE &HE139,16
1480 POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$
1490 CONSOLE 0,24:IF PEEK(&HD07F) THEN KLIST 1 ELSE POKE &HE139,dirg
1500 proces=proces-1:CHAIN proces(proces)
1510 '------( OPEN )-----
1520 '
1530 POKE v_ddrv+1,1,7:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1540 POKE v_od,2:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1550 m=PEEK(v_mac):IF PEEK(v_stop) RETURN
1560 POKE v_sbdr,1:POKE v_op,3:d$=USR1(fe$):RETURN
1570 '------( ERROR ROUTINE )-----
1580 '
1590 LABEL "!3":RESTORE "m2":GOTO 1620
1600 LABEL "!2":RESTORE "m1":GOTO 1620
1610 LABEL "!" :RESTORE "m0"
1620 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1630 POKE v_stop,0:GOTO 1440
1640 '------( DATA AREA )-----
1650 LABEL "m0":DATA エラーが発生しました!!
1660 'LABEL "m0":DATA Error !!
1670 LABEL "m1":DATA ファイル・ネームを指定してください
1680 'LABEL "m1":DATA Need FILE-NAME
1690 LABEL "m2":DATA リポートできません
1700 'LABEL "m2":DATA Not REBOOT

```

ットのディレクトリを作ります。ユーザーはフォーマットの違いを意識する必要はありません。これを使えば「X68000のディスクをX1turboZで編集する」といったことも可能です。

グラフィックローダ/セーバ

グラフィック特集にあわせて、画面のロード/セーブを行うプログラムをKAME-DOS上で開発しました。特集のほうのプログラムはturboZオンリーですが、このローダとセーバはX1シリーズ全機種で使用可能です。Z-BASIC以外の標準BASICにはこのような命令がなかったので、X1間での画像のやりとりも多少便利になると思います。詳しい説明は特集記事に譲るので、ここではその紹介だけしておきましょう。

命令: GLOAD

書式: GLOAD ファイルネーム

プログラム: 特集を参照

使い方はほかの外部コマンドと同じです。

特集のプログラムから子プロセス的に呼び出されるので、通常の外部コマンドとは少し異なります。

命令: GSAVE

書式: GSAVE ファイルネーム

プログラム: 特集を参照

GLOADと同じ。

解説! プログラミング

今月は短くて、しかもBASICプログラムなので難しいことはありません。ですから「外部コマンドの作成作法について」を中心に展開してみましょう。

第1部 起動

まず、コマンドラインからコマンド名が打ち込まれました。COMMAND.X1はそれが内部コマンドではないと判断して、ドライブにコマンドと同じファイル名を探しにいきます。なければエラーで、あれば拡張子が「.X1」かどうか(外部コマンドかどうか)を見て処理を振り分けます(図3)。

外部コマンドならCHAINして、そうじゃなければRUNします。ここが重要で、CHAINによってCOMMAND.X1で定義された変数があるまま引き継がれます。外部コマンド側では必要に応じてその変数を

図3 外部コマンドの動作



使うことになります。だから、実行中にプログラムを止めて再実行することができないのです。これは必要な変数を何度も定義しないようにして、外部コマンド側の負担を軽くするためです。

それならば、ここでいう必要な変数とはなんでしょう？ 内部・外部に関わらずコマンドを実行するときには、KAME-DOS共通のD000_H番地以降に常駐しているマシン語プログラムをアクセスします（7月号のアセンブルリストのこと）。マシン語オンリーで開発しているのならアドレスはラベルに固定できますが、BASICによる開発だとアドレスを変数に定義して、ラベルとして使う必要があります。いわばこれらはグローバル変数で、コマンド内でのみ使われるのがローカル変数というところです。なお、COMMAND.X1をリブートすると一度すべての変数をクリアするので、使用変数がたまりすぎることはありません。

第2部 実行

外部コマンドはその利用目的によって相当異なった作りになるので、一言ではいいきれないものがあります。ただ、大きく分けると次の3つになります。

- 1) ファイルを扱うコマンド
- 2) ディスクを扱うコマンド
- 3) それ以外のコマンド

1)は、主に内部コマンドに採用されているものでCOPYやDIRなどになります（もちろんCOPYと同じことをする外部コマンドを作ることも可能です）。特徴として、ファイルをアクセスする前には必ずそのファ

イルをOPENし、書き込んだあとにはCLOSEするということです。そのためOPEN/CLOSEルーチン呼び出す必要があります。また、実際にファイルの中身をアクセスするルーチンも使われるでしょう。開発する場合は、一番面倒なコマンドになります。

2)はFORMATやDISKCOPYのことです（今は発表できませんが、そのうちに発表したいと思います）。これらはディレクトリとかFATとか、ランダムアクセスの部分がいらないので比較的簡単に開発できます。マシン語ルーチンも低級な（ハードを直接アクセスする）ルーチンを使うのでわかりやすくなります。

3)は、特にKAME-DOS上で開発する必要はないようなプログラムです。ご存じのとおりKAME-DOSはディスクアクセスルーチンの集合体です。そのうえでディスクを使わないようなプログラムを動かしても、マシン語の常駐部分だけメモリの無駄になります。開発環境も整備されていないので、このようなプログラムは発表しないつもりです。

今月のMD,RD,GLOAD,GSAVEは1)にあたります。MD,RDの解説で、外部コマンドの雰囲気をつかんでください。

第3部 リブート

COMMAND.X1へリブートするときには、それ専用のマシン語ルーチンが用意されています。注意点は前に書いてあるとおりです。GLOAD,GSAVEに関してはこのルーチンを使わずに単なるCHAIN命令で

済ませています。普通はリブートルーチンを使います。最終的にはCHAINを使うので、COMMAND.X1に戻ったときの結果は同じです。

これは別にCOMMAND.X1へのリブートだけじゃなくて、「PROCESS」という変数で管理されているひとつ上の親プロセスへ戻るために使います。つまり、COMMAND.X1というのは親プロセスというかたちになっています。

MDとRD

どちらのプログラムでもまず、DEFUSRを定義しています。これらは先に解説したマシン語ルーチンのアドレスです。変数の頭に「M,V,S」がついているのはCOMMAND.X1からの持ち越し変数です。次に、いわゆる「OPEN」と「ディレクトリ名が指定してあるかどうか」のチェックをし

MS-DOSのディレクトリ

MS-DOSフォーマットのディレクトリ管理方法は、X1のそれとちょっと違っています。下位ディレクトリの先頭には、「.」「..」というファイル名が2つ記録されています。これはMS-DOSでは「カレントディレクトリ」と「親ディレクトリ」を表していて、そのクラスタ番号も記録してあります。実際にこれらを使って管理しているかどうかはわかりませんが、X1にはこれに相当するものはありません。そこで、KAME-DOSでは上記のようなファイル名が出てきたら無視を決め込みます。親ディレクトリのクラスタ番号は、内部ワークエリアに保存しておくようにしました。

リスト3 MD.X1

```

1000 'MD (MKDIR) ver 1.0 By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=&HEE80:DEFUSR3=m_tranr
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 POKE v_od,1:GOSUB 1260:IF PEEK(v_stop) THEN "erre"
1060 IF fe$(1)="" OR fe$(1)="/" THEN "errx"
1070 ON PEEK(v_mac) GOSUB 1370,1350,1370,1390
1080 GOSUB 1450
1090 POKE v_edw,k:POKE v_zoku+1,i1:MEM$(v_msbt,2)=MKI$(i0)
1100 MEM$(v_bf,2)=MKI$(buff):POKE v_frwf,1:CALL m_crsw
1110 POKE v_frwf,0:IF PEEK(v_stop) THEN "erre"
1120 CALL m_saved:IF PEEK(v_stop) THEN "erre"
1130 d$=USR3(process$(proces-1)):IF PEEK(v_stop) THEN "errb"
1140 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+":"+RIGHT$(STR$(3-PEEK(v_mac)),1)
1150 proces=proces-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1160 '----- ERROR
1170 '
1180 LABEL "erre":RESTORE "d1":GOTO 1200
1190 LABEL "errx":RESTORE "d2"
1200 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:POKE v_stop,0:GOTO 1130
1210 LABEL "errb":RESTORE "reb"
1220 READ m$:PRINT:CREV 1:PRINT m$;:CREV 0:PRINT:d$=INKEY$(1)
1230 POKE v_stop,0:GOTO 1130
1240 '----- SUB
1250 '
1260 LABEL "open"
1270 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1280 fe$(1)=RIGHT$(fe$(1),PEEK(v_yen)):IF fe$(1)="" OR fe$(1)="/" RETURN
1290 k=PEEK(v_mac):d=INSTR(fe$(1),".")
  
```



```

1300 IF (k=1 OR k=3) AND d=0 THEN fe$(1)=fe$(1)+".DIR"
1310 POKE v_sbdr,2:POKE v_op,3:d$=USR0(fe$(1))
1320 MEM$(v_fs1,4)=CHR$(0,0,0,0):MEM$(v_fnam1+46+22,5)=CHR$(0,0,0,0,0)
1330 RETURN
1340 '
1350 LABEL "ms"
1360 k=1:i0=1024:i1=&H10:d=0 :RETURN
1370 LABEL "x1"
1380 k=1:i0=256 :i1=&HC0:d=&HFF:RETURN
1390 LABEL "m2"
1400 k=2:i0=1024:i1=&H10:d=0 :RETURN
1410 '
1420 LABEL "poke"
1430 d$=USR2(MKI$(p)+MKI$(1)+CHR$(j)):p=p+1:RETURN
1440 '----- DATA
1450 LABEL "mem"
1460 MEM$(&HEE80,16)=HEXCHR$( "EB 5E 23 56 23 4E 23 46 23 CD 93 EE 13 0B 78 B1" )
1470 MEM$(&HEE90,16)=HEXCHR$( "20 F7 C9 7E C3 27 E0 00 00 00 00 00 00 00 00" )
1480 MEM$(&HEE95,2)=MKI$(m_lddea)
1490 d$=USR2(MKI$(buff)+MKI$(10)+CHR$(d))
1500 i=PEEK(v_mac):IF i=1 OR i=3 RETURN
1510 '----- for MS-DOS
1520 RESTORE 1600
1530 p=buff :FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1540 p=buff+32:FOR i=0 TO 11:READ j:GOSUB "poke":NEXT
1550 p=buff+26:j=PEEK(v_crs):GOSUB "poke":j=PEEK(v_crs+1):GOSUB "poke"
1560 p=buff+58:i=PEEK(v_csdire)+PEEK(v_dn):IF i=0 THEN 1590
1570 h=v_csdire+26+8*PEEK(v_dn)+2*(i-1)
1580 j=PEEK(h):GOSUB "poke":j=PEEK(h+1):GOSUB "poke":RETURN
1590 j=0:GOSUB "poke":j=0:GOSUB "poke":RETURN
1600 '
1610 DATA 46,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
1620 DATA 46,46,32,32,32,32,32,32,32,32,32,32,32,32,32,32
1630 '----- MESSAGE
1640 '
1650 LABEL "d1":DATA エラーが発生しました!!
1660 'LABEL "d1":DATA Error !!
1670 LABEL "d2":DATA ファイル名を指定して実行してください
1680 'LABEL "d2":DATA What file-name?
1690 LABEL "reb":DATA リブートできません
1700 'LABEL "reb":DATA reboot error

```

リスト4 RD.X1

```

1000 'RD (RMDIR) ver 1.0 By M.Kameda
1010 '
1020 DEFUSR0=m_opens:DEFUSR1=m_preop:DEFUSR2=m_tranr
1030 '
1040 POKE v_dn,PEEK(s_dn):POKE v_mac,PEEK(s_mac4+PEEK(s_dn))
1050 IF fe$(1)=" THEN "errx"
1060 POKE v_od,1:GOSUB 1270:k=PEEK(v_stop):POKE v_stop,0
1070 IF k=0 THEN "erry" ELSE IF k<3 THEN "erre"
1080 GOSUB 1320:IF fe$(1)=" THEN "errx"
1090 k=PEEK(v_stop):IF k=3 THEN "errz" ELSE IF k THEN "erre"
1100 CALL m_diflat:CALL m_dldir:CALL m_clos2
1110 '
1120 d$=USR2(process-(process-1)):IF PEEK(v_stop) THEN "errb"
1130 k=PEEK(v_dn):IF k<4 THEN DEVICE STR$(k)+": "+RIGHT$(STR$(3-PEEK(v_mac)),1)
1140 process=process-1:CHAIN MEM$(v_p256+&H81,PEEK(v_p256+&H80))
1150 '----- ERROR
1160 '
1170 LABEL "erre":RESTORE "d1":GOTO 1210
1180 LABEL "errx":RESTORE "d2":GOTO 1210
1190 LABEL "erry":RESTORE "d3":GOTO 1210
1200 LABEL "errz":RESTORE "d4"
1210 READ m$:PRINT:CREV 1:PRINT m$::CREV 0:PRINT:POKE v_stop,0:GOTO 1120
1220 LABEL "errb":RESTORE "reb"
1230 READ m$:PRINT:CREV 1:PRINT m$::CREV 0:PRINT:d$=INKEY$(1)
1240 POKE v_stop,0:GOTO 1120
1250 '----- SUB
1260 '
1270 LABEL "open"
1280 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1290 fe$=RIGHT$(fe$(1),PEEK(v_yen))
1300 POKE v_sbdr,0:POKE v_op,1:d$=USR0(fe$):RETURN
1310 '
1320 LABEL "dopen"
1330 i=INSTR(fe$(1),"/"):IF i THEN 1380
1340 d$=USR1(fe$(1)):IF PEEK(v_stop) THEN RETURN
1350 k=PEEK(v_dn):j=PEEK(v_csdire+k):POKE v_fnam1+46+43,j
1360 i=v_csdire+26+k*8+(j-1)*2:POKE v_fnam1+46+44,PEEK(i),PEEK(i+1)
1370 POKE v_sbdr,2:POKE v_op,2:d$=USR0(fe$(1)):RETURN
1380 '
1390 fe$="":WHILE i
1400 fe$=fe$+LEFT$(fe$(1),i):fe$(1)=RIGHT$(fe$(1),LEN(fe$(1))-i)
1410 i=INSTR(fe$(1),"/")
1420 WEND:IF fe$(1)=" THEN RETURN
1430 d$=USR1(fe$):IF PEEK(v_stop) THEN RETURN
1440 GOTO 1370
1450 '----- MESSAGE
1460 '
1470 LABEL "d1":DATA エラーが発生しました!!
1480 'LABEL "d1":DATA Error !!
1490 LABEL "d2":DATA ファイル名を指定して実行してください
1500 'LABEL "d2":DATA What file-name?
1510 LABEL "d3":DATA ディレクトリにファイルがあります
1520 'LABEL "d3":DATA File exists
1530 LABEL "d4":DATA 指定されたディレクトリがありません
1540 'LABEL "d4":DATA No directory
1550 LABEL "reb":DATA リブートできません
1560 'LABEL "reb":DATA reboot error

```

す。エラーは1カ所にまとめて同一の処理がなされます。

MDではMD独自のマシン語プログラムを持っています。これはディレクトリ領域初期化の高速化のためです。そして、このように短いマシン語プログラムを使う場合は、EE00番地からの256バイトを使うことになります。ここは汎用ワークエリアなので保存はしておけませんが、一時的に置いておくことはできます（ほかの外部プログラムでもこうしていくつもありです）。

その後ろにはMS-DOS用の特別初期化ルーチンが続いています。実際の書き込みは「mcrrsw」ルーチンをコールすることで行われます。そして、エラーがなければ「msaved」ルーチンでいま書き込んだディレクトリをCLOSEします。これらのルーチンは、ただコールしただけじゃ正常には動きません。その前後で盛んにPOKEしているように、あらかじめ値を設定しておかなければならないのです。POKEアドレスの意味は7月号のアセンブルリストを見ればわかるでしょう。

そして最後はリブートルーチンです。USR3命令からCHAIN命令までがそうで、これはRDでも同じです。USR3で親プロセスを引数にして、その結果はVP256+&H81からに格納されています。この内容はフルパスファイルネームです。ディスクが入れ替えられている可能性も考慮して、DEVICE命令も実行しています。

RDでも基本的な作りは同じですが、OPENとCLOSEの部分が違っています。OPENが2つに分かれているのは、「ディレクトリ自体のOPEN」と「そのディレクトリ内にファイルがあるかどうかを調べるOPEN」があるからです。CLOSEの場合は、ファイルを消すのとわけが違って、3回に分けたコールが必要になります。そのほかには、初期化する必要もないのでMDのようなマシン語ルーチンはありません。

これで外部コマンドの概要はわかってもらえたと思います。まだ作るにはいたらないかもしれませんが、わかるところを改造してみるのもいいでしょう。来月はずっと突っ込んだ説明をして、なにか新しいコマンドを発表しながら、実際にコマンドが作れるようになるくらいまではやりたいと思っています。

対戦ポピュラス 祝一平VS西川善司

実況・解説 浦川博之

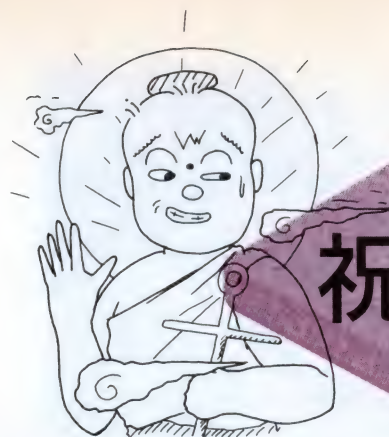


illustration: I.Yamada

編集室で対戦ポピュラスなんかやられちゃ面白くなって大迷惑。なのに西川君が祝氏に挑戦状をFAXで送っちゃうんだから、さあ大変。100号記念なのに、もっと実のある企画はないのかあ〜、といいつつOh!X史上最大の決戦の火蓋は切って落とされた。

5月上旬のある日。そもそも編集室にはX68000が2台並んでいるのが悪い。これで対戦ポピュラスをやるなったって無理というもの。かくして今日もスタッフの対戦が行われるわけです。なかでもズバ抜けて強いのが西川善司。270面を制覇し、対戦は負けたことがないとか。

善「まあ、ぼくにかなう人はいないかな」
編「いや、祝さんがAmigaで始めて、いま420面だからわかりませんよ。ね、祝さん」
善「フフフ、負けませんよ、祝さん」
祝「(ニヤッと笑って中指を立てる)」

すでにこの会話以来、2人の対決は必然だったのです。

*

5/28 18:30決闘当日。

善「来た、祝さん」

祝「……いたな、青二才めが」

善「ひょっとしてあのFAX、怒ってる？」

祝「叩き潰してくれる」

おおっと、出会い頭にこのエキサイトぶり。おや、観客の中に丹明彦さんの姿が。丹さん、丹さんは善司くんを負かす寸前まで追いこんだそうですね。

丹「ええ、向こうが何もできなくなるところまでいったんですが、いつの間にか逆転されてしまいました。はは」

お、祝氏が自分のマウスと専用マット(なぜか航空機力学の本)を持って現れた。

善「道具まで気にしちゃって、もう」

と言いつつ、善司くんもマット代わりのフロッピーケースを取りに戻っている。

さて、今回の対戦のマップはレビューを書いた中野修一氏が作った特製だということとす。中野さん、ちょっとすいません、

どんなマップか教えていただけます？」

中「ええ、いろいろあります」

へえ、たとえば？」

中「(ニヤリと笑って)結構スゴイです」

うーん、この人も意味不明な気合いが入ってるな。さて……。

中「どのマップにします？」

祝「じゃあ、この砂漠のにしよう。異存はないな？」

善「どのマップでも同じですよ。へへ」

おおおとギャラリーが沸く。

ここでちょっとマップの説明を。地形は完全に対称で、お互い人口は1人ずつでスタート。奇跡はすべて起こせます。ひとつ変わっているのは、沼が「底無し」に設定されている点。普通は1人沼に落ちるとそのマスは平地に戻るんだけど、このマップではいつまでたっても人が落ち続けるというわけ。沼を作られたら最優先で直さないとマズいわけですな。

20:53 さあ、ゲームスタート。

祝「あれ、人はどこにいるんだ？」

中「(ニヤリと笑って)え、いるじゃないですか」

どどーん。中央にそれらしく島を作っておきながら、人はマップの隅に、しかも岩に囲まれて細々とテントを立てていた。これじゃあ思うように家を増やせない。確かにスゴいマップだ。

善「あーっ、ちくしょう。でいだい」

あああ、むりやりテントを城にする気か。中野氏が「あっあっあっ」と心配そうにも嬉しそうな悲鳴をあげる。

家を作らせてもらえない悪魔の民。ふらふらとさまよっているうちに……。

YOU LOST

SOCRE 2570

あーっ、なんと開始後1分で西川善司の連勝記録ストップ! あまりの情けなさにギャラリーは開いた口がふさがらない。丹「体力もないのに砂漠を歩かせたりするから……」

祝「うっはっは、口ほどにもない」

善「しまった。気にしすぎたあ」

中「だからあ、もうちょっと気合い入れませんか？」

狙いどおりの展開に嬉しそうな中野氏。

気合いを入れ直して、20:55再開。今度は2人とも慎重に人が増えるのを待っているようです。

祝「死ぬなよ死ぬなよ……よおーし!」

城は小さい家に比べて人の増えるペースが速いが、収容人員が多いため人があふれるのには時間がかかる。ということは、「城をときどき壊して人を追い出す」というのが常用テクニックになるわけです。2人は次々と城を作っては追い出し平地を開拓しています。左上のマップを見ていると、平たい大陸がじわじわ中央に向かって伸びていくのがブキミ。

スゴゴゴゴゴ。おおーっと、祝氏の領土に地震。最初にしかけたのは善司くんだ。

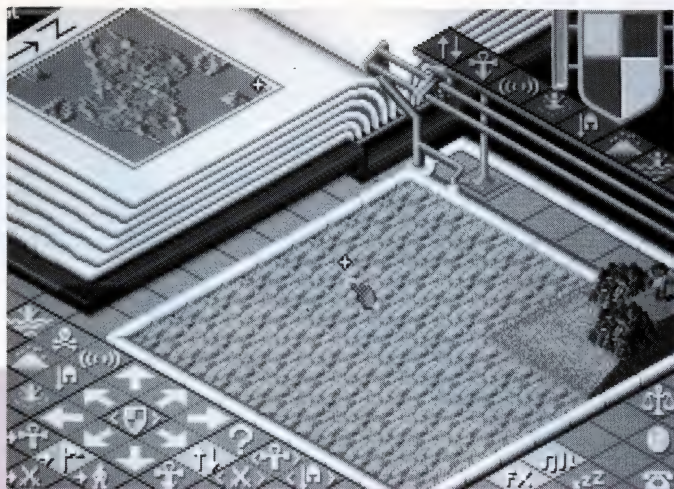
祝「地震なんか効かないもん」

さっさと修復してしまう祝氏。マウスのクリックにムダがない。さすが420面はダテじゃないぞ。

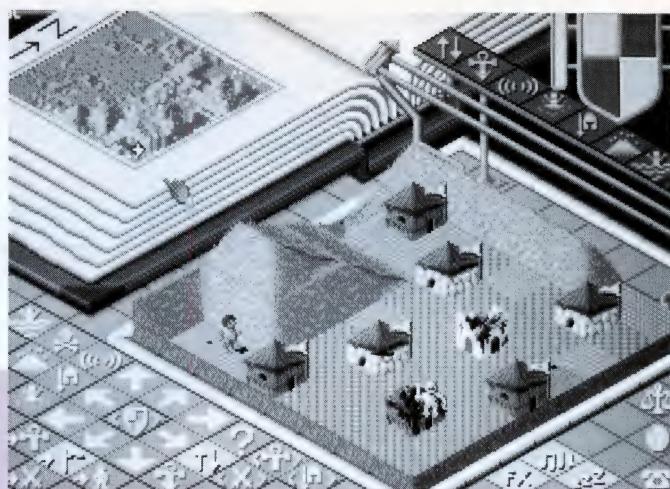
Oh!X通巻100号に寄せて

ども。ポピュラスやりたさにパソコンを買ってしまった浦川です。いま100面ちょっとですけどね。え? 機種ですか? よ、弱ったな、なんでもいいじゃないですか。

まあ、自宅でじっくりとコンピュータをイジめるのもいいですが、人間相手となるとまた格別。うひひ。Oh!X編集部にはたくさんX68000がありまして(当たり前だ)、その中には2台RS-232Cでつながれたヤツもあります。これを見ると、さもX68000が「ぼくたち対戦ポピュラスのためにここにいまーす」と言っているような気がして、つついそこら辺の人に「やろーやろー」と声をかけてしまう。かくして編集者がドアを開けて入ってくるなり、締め切り間際のライターがマウスをカチカチやっている姿に頭をかかえるという日々が続くのでした。おっと、これのどきが100号記念の祝辞なんだろう。というわけで合掌。



これが1回戦のマップの初期状態。左右対称の大陸と丹念に岩が配置されている。ところどころに沼も置いてあるという。ちなみにリーダーはいない。



戦況はかなり煮詰ってきた。かつての海はどこへやら……。下か祝一平，上が西川善司。手作りの山攻撃が城や町を破壊している。もう泥沼。

ゴボシ あれ？

ゴボシ 何の音だ？

ゴボシ ぬ、沼だ！

だのははと無責任に笑うギャラリー（なぜか驚嘆より笑いが先に立ってしまう）。

祝「やりおったな」

善「いえ、なんにもしてませんよお」

祝「と、いうことは……」

ギャラリー全員の視線が中野氏に集まる。

善「まさか、最初っからあるんじゃないか」

中「まあ見てのお楽しみ」

やはり中央の島に沼があった。新大陸に家を建てようと勇んで出かけた民は、この沼に沈んでいたわけ。中野氏恐るべし。

祝「うーん、砂漠の沼は発見しづらい」

善「中野さん、あらかじめ設定しとくなんてすごいイジワル（スゴゴゴゴ）」

しっかりスキを見て地震を起こす善司くん。

丹「砂漠で地震は効きますよ」

へえ、なんで？

丹「ほら、外をうろちょろしてる間に体力がなくなっちゃうから」

善司くんはマナが貯まるたびに地震をしかけます。対照的に祝氏はマナを貯めながらひたすら領土を拡大。

21:35 ふたりの領土がそろそろ接してきました。善司くんが一番敵地に近い家を探して、画面の端にくるように設定している。

カチカチカチカチ……

みるみる相手の土地が盛り上がる。ワッハッハと無責任に笑うギャラリー。でた。これが善司くんの得意技、手作りの山だ。

祝氏は地震で素早く取り壊す。しかし修復し終えたところにはマップのほかの場所でずんずんと巨大なピラミッドが立っている。

祝「むう（シュイイン）」

おーっと、怒った祝氏が火山をお見舞いだー！ しかも二段重ね！

*

22:00 あれから1時間。手作りの山と火山が乱れ飛んで、かつての平地はどこへやら。家の数を見ると善司くんのほうが押し気味ではあるけど、人口ゲージを見るとまだまだ互角。人数が多いので次第に処理速度も落ちてきた。しかもハングアップ防止のため2400ポでやっているのでおさら。マウスの反応が悪くてときどきヘンなところがぼこっと盛り上がりつつあります。

22:30 開始から1時間半たって戦いはやや膠着状態に。そろそろ休憩にしません？

祝「向こうが泣いて頼むんだったら休んでやってもいいよ」

善「もう、祝さんったら強情なんだから。

素直に休みたいと言えいいのに」

祝「なに、そんなに休みたいの？」

善「まさか。祝さんが泣いて頼むんだったらべつですけど」

次第に善司くんがじわじわと平地を獲得している。やはり手作りの山の対応に追われ続けている祝氏の不利は否めない。ところで祝さんが手作りの山はほとんどしかないのは、なにか信条があるのだろうか？

祝「おい、休んでやってもいいよ」

善「いいですよ、べつに」

祝「……休んでやってもいいんだよ」

善「だからいいってば（フォン）」

ああっと、騎士が誕生。対戦ではよほど有利でないとできない行為だ。散在する祝氏の家を焼いてまわる騎士。さらに手作りの山攻撃が襲いかかる。これらを全部修復しながら挽回をはかるのは祝氏といえども至難の技だ。

祝「むっ。くそっ。くそっ」

脂汗をにじませながら力をこめてクリックを続ける祝氏。反応が鈍いんだから、そんなに力をこめたって……。

祝「うるさい。やってるほうの身にもなってみろ」

ついにいっぱいだった人口ゲージも減少を始めた。騎士が次から次へと送りこまれ、あっちこっちで山が立つ。祝氏側の家は端のほうに散在するばかり。

そして23:07。

祝「……うむ。今日のところは負けにしていあげよう」

ついに祝氏敗北宣言！ 西川善司のTKO勝ちで決着！

祝さん、敗因は？

祝「若さに負けた」

2時間20分の長丁場ですからね。

祝「それから、あの沼は発見しづらいからキライ。そもそもマップを作ったあのコミッショナーが悪い」

勝った善司くんは？

善「そうねえ、へへへ。まあ、丹さんのほうが強かったかな。なんちて。ぽっくん」

祝「この借りは必ず返すぞ」

善「いつでも来なさい。はっはっは」

*

その4日後。

「ちわーす」編集室に入っていくと、さっそく再戦している2人の姿があった。

善「祝さんが泣いて頼むからさあ」

祝「この前のは練習。今度が本番」

2人とも好きにしていようだい。

今度はもっと素直なマップで対戦。雪原に点対称に日本が2つ配置され、沖縄に1人だけ人間がいるという設定です。

おや、祝氏が家をくずして、一番低い平地で展開するのに対して、善司くんは一段

高いところで展開している。洪水対策か？
丹（また見に来ている）「いや、やりこんだ人なら洪水は使いません。火山を何発も起こしたほうが有効ですから」

高い土地をいじるほうがマナがいるんですよね。マナの少ない序盤にこういうことをしていいのかなあ。

19:45 やはり人口比7:3ぐらいに差がついて、今度は祝氏が中央部を押さえた。苦しい善司くん手作りの山で反撃！ また泥沼の戦いが始まる。立てる崩す、立てる崩す、立てる崩す、沼にはまる。

善「やっぱり沼が奇跡のなかでは一番有効ですからね」

祝「えっ？ 沼の弱点知らないの？」

善「……そんなこと言って動揺を誘おうとしてるんでしょ」

祝「そう思う？」

直接対戦ならではの口頭の戦い。

20:17 祝氏がメガネをはずした。気合いの入れ直し（どうでもいいが、氏はサングラスがとても似合うお方である）？

お互いの境界にまんべんなく山が立っている。やはり山の被害のせいか、祝氏のリード幅が縮んだような。

「**シュイイイン**」あ、火山だ、祝氏が火山をおみまい！ さらにシンボルを移動にかかると……。ここで一気に攻勢に出るのか。

祝「あれ、できない」

リーダーは敵陣との境で死んでいた（笑）。

善司くんは山を作って、相手の復旧の間に領地を広げる作戦に、祝氏はリーダーを誘導して個別撃破の作戦に出ています。

20:33 祝氏のリーダーは合体を繰り返し、パワーのある奴になりました。楽しげに誘導先を選ぶ祝氏ですがその途端……。

ゴボシ

祝「……！」

リーダーのいたところには沼が広がっていた。ギャラリーが無責任に笑う。

祝「……（**シュイイイン**）」

善司くんの領土に怒りの火山が炸裂！

21:10 そろそろ勢力が五分五分というところ。やはり善司くんは攻勢にたけています。おっと、何を考えたか善司くんが自分の領土に地震をしかけました。

善「こうやってシンボルに人を集めるんですよ」

恐るべき早さで最強の騎士が誕生。さらに騎士が敵地に向かっている間にも手加減しない善司くん。

善「ああ、祝さんたら僕に無断でこんなところに城を（**カチカチカチカチ**）」

山を立てている間に騎士が祝氏の領土に到着！ が、祝氏は慌てずに騎士の周りに穴を掘り、騎士を水の中に沈めてしまったあ。もがく騎士。体力が少しずつ落ち始める。善司くんはぜんぜん気がついていない。ギャラリーは笑いたいのを必死にこらえています。そのまま何事もなかったかのように自分の領土を整備している祝氏。数分してふと善司くんが右上のウィンドウを見ると……。

善「ああっ、なんかもがいてるう」

だははははと爆笑するギャラリー。たちまち敵住民を池に落とすという「水攻め攻撃」が乱れ飛びました。

22:27 山を残しながら、自分の領地はしっかりキープしている2人。しかしやはり中心部は善司くんが取り、祝氏は周辺部に追われています。自分の領地に地震をしかけている善司くん。出てきた人間を、シンボルのある敵陣まっただなかに集合させる

「一方的ハルマゲドン」攻撃です。騎士同様の追い込み技ですね。

しかしそれでも事態は終結しない。千日戦争状態にあると判断した中野氏が、善司くんにハルマゲドンを起こすよう指導勧告。以後善司くんは奇跡を起こすのを控え目にして、マナの集積をはかる。一方祝氏は、再びマウスを汗だくでクリック。

祝「もーいや、こんな生活」

挽回はできなかったが、この抵抗が効いて善司くんがハルマゲドンを起こすまでにはさらに1時間を要したのだった。

23:45 「**ウホウホウホ**」ハルマゲドンスタート。人口ゲージは祝氏の圧倒的不利を伝えている。ああ、やはり祝氏も善司くんの独走を止められなかったか。画面の中で2人のリーダーが向き合った瞬間！

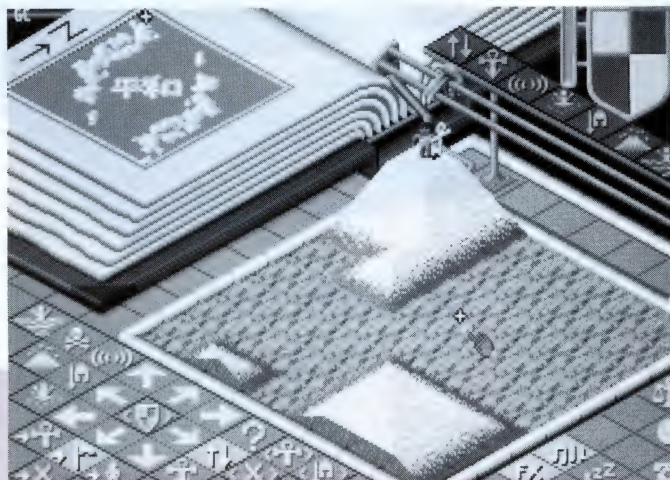
ピタ！

うおお、ハングだあ！ 天は祝氏に武士の情けをかけようというのかー！

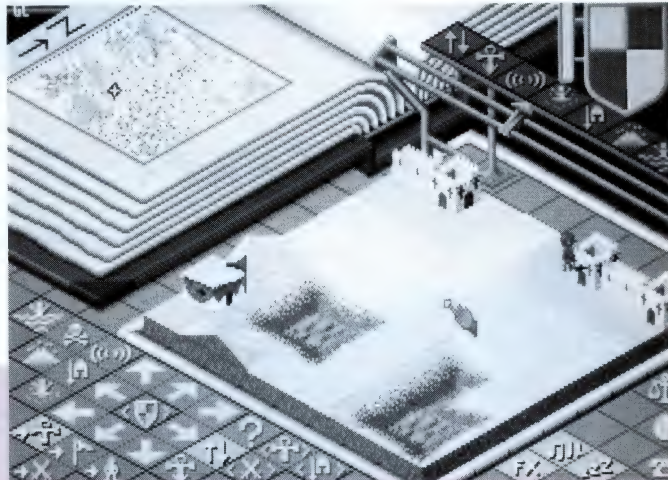
結局波乱のラストを乗り越えて、西川善司のハルマゲドン勝ちが決定しました。

結局善司くんの2勝という形になりましたが、聞いたところでは祝氏は対戦がこれで3回目ということですから、いかに420面まで進んでいても、対戦ポピュラスのノウハウのある西川善司くんに一日の長があったといえるでしょう。

しかし、この対戦もさらなる戦慄の歴史の序章に過ぎないのです。このあともさらに西川善司対中野修一などの数々の恐ろしい戦いが、編集室では繰り広げられています。対戦ポピュラスは確かに面白い。時間は使うし電話代もかかるし友人関係も下手するとこわれる。それでも対戦ポピュラスは面白い。あなたはこの面白さにつかってみる勇気がありますか？



2回戦。中野氏による平和島マップ。祝氏のリクエストで気候は氷河時代となった。今度はなんの仕掛けもない。赤い敵が樺太から……（ちょっとあぶない）。



下が西川氏で上が祝氏。画面上のあちこちにボツボツと穴が見える。ちまたでは「温泉」と呼ばれている。善司くんの地震突撃攻撃対祝氏の執拗な沼攻撃。

X68000 10万台突破記念

愛読者特大 モニタープレゼント

Oh!Xは通巻100号なんだよ～、とはしゃいでいたら、ほとんど時期を同じくしてわれらがX68000が10万台出荷を達成した。これぞ歓喜の2段重ね！ここはひとつシャープさんにお願い！というわけで豪華プレゼントを提供していただきました。どうです、スゴいでしょ。特に大型ディスプレイやカラーイメージスキャナなんて持っている人、少ないんじゃないかな。えっ、本体はないのかって？だって大部分の皆さんはすでにX68000ユーザーじゃないですか。それに周辺機器ならX1/turboユーザーでも使えるでしょ。なに、X68000に乗り換えたい？だったら本体ぐらい自分で買わなきゃね（というのがOh!Xの本音なのだ）。なお、9番以外はモニタープレゼントだから、当たった人には感想文をお願いします。

1



21型カラーディスプレイ

CU-21HD 148,000円 1名

着脱可能なスピーカーを搭載した大きなカラーディスプレイ。これでゲームをやったらさぞかし気持ちいいことでしょう。

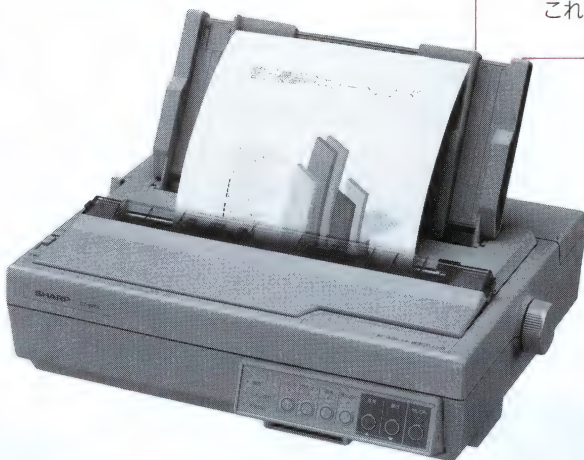
2 熱転写カラー漢字プリンタ

CZ-8PC4

99,800円

1名

48ドット、7色のカラー印字ができるプリンタ。もちろんグラフィックもプリントできるぞ。いろいろなカラーリボンも使える。



3

カラーイメージ スキャナ



CZ-8NS1 188,000円 1名

最大A4サイズの絵や写真をフルカラーで読み込むことができるカラーイメージスキャナだ。

4

サイバースティック

CZ-8NJ2

23,800円

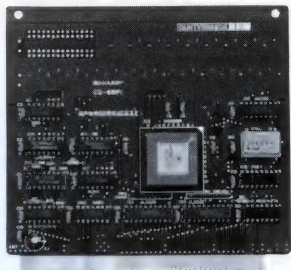
1名



ゲーム命の人ならば、ぜひ手にいれてほしいアナログジョイスティック。細かな操作も行いやすくなるぞ。

5

数値演算 プロセッサ ボード



CZ-6BP1 79,800円 1名

面倒な計算やレイトレーシング、シェーディングなどの処理速度を一気に高めることができるこのボード、CGには最適。

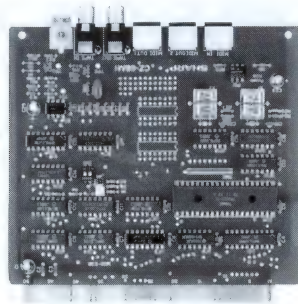
6

MIDIボード

CZ-6BM1

26,800円

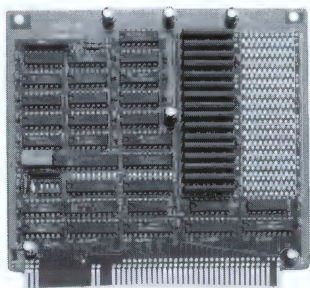
1名



最近はいろいろなゲームもMIDI対応になっている。このボードがあればMIDI楽器が接続でき、鮮やかなサウンドが楽しめる。

7

2MB増設RAMボード



CZ-6BE2 79,800円 1名

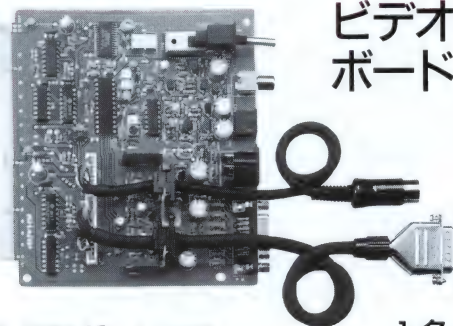
あれもこれもパソコンでやりたい、という人はRAMボードの増設は必至。そんなあなたにこのボードをプレゼント。
(1M増設済のこと)

プレゼントの応募方法

とじ込みのアンケートはがき(ただし、今月のもの)の該当項目をすべてご記入のうえ、希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年8月18日の到着分までとします。当選者の発表は1990年10月号で行います。

8

ビデオ ボード



CZ-6BV1 21,000円

1名

このボードを使えば、X68000で作ったグラフィックや、プレイしているゲームなどが、簡単にビデオに録画できるようになるぞ。

(以上、シャープ提供)

9

キーボード延長ケーブル

1,980円



黒/グレイ
各5名

九十九電機より創刊100号を記念して、オリジナルのキーボード延長コードをプレゼント。寝っ転がってキーボードも打てるかな。

6月号プレゼント当選者

1A)ジェミニウイング(沖縄県) 宇良秀樹 B)闇の血族(静岡県) 野村一洋 2ねじ式(千葉県) 田浦達也(山口県) 大隅研治 3The File Professor(東京都) 高橋信博(静岡県) 戸塚昭信 4サイクロンEXPRESSα(秋田県) 佐々木仁志(神奈川県) 鈴木利明(東京都) 三田恭一郎(静岡県) 三橋和美(大阪府) 藤沢直樹 5A)FAR SIDE MOON(広島県) 本谷正樹(愛媛県) 横山智生 B)A列車で行こうII(富山県) 加賀見政和(長崎県) 佐藤充浩 C)大海令(埼玉県) 桑原智志(岡山県) 梅田敬 D)南海の死闘(東京都) 小山薫(広島県) 岸本秀生 6クオース(北海道) 加納一郎(福島県) 村上健(京都府) 村久木康夫 7ジャック・ニクラウス・テレフォンカード(宮城県) 伊藤洋美(東京都) 平尾雄一(神奈川県) 長嶺隆(奈良県) 野瀬正博 林衛 8スタークルーザー X68000用(福島県) 岩瀬正樹(東京都) 大橋飛雄吾 Xlturbo用(神奈川県) 田口聡(岡山県) 小谷恒 9キューブランナー(東京都) 角野俊人(神奈川県) 武藤俊哉(京都府) 田中啓 10レナム(岩手県) 片岸健一(群馬県) 石山篤志(兵庫県) 郡茂樹(新潟県) 霜島博史(香川県) 佐竹勝博 11A)ガンマ・プラネット(東京都) 高橋明(群馬県) 藤田明(愛知県) 永井周作 B)グランディフロム(千葉県) 久原義弘(栃木県) 佐藤崇(三重県) 大橋隆太郎 C)Simple-CAD X68K(福島県) 仲山秀樹(和歌山県) 辻本浩一 12上海II(長野県) 吉沢克明(兵庫県) 堀江良孝 13ボピュラス(千葉県) 佐藤一成(島根県) 原誠(鹿児島県) 園田光太郎 14プログラムオペレーティングシステム(東京都) 木部幸雄(石川県) 川口聡 15PIO-6BE1-A(東京都) 飯塚晃太郎 16銀河英雄伝説+set(埼玉県) 武藤一文 加藤勲(京都府) 牧本隆 17G68K II(東京都) 信川洋(福岡県) 平山謙司(宮城県) 土井順之 18A)D-RETURN(神奈川県) 細井実人(茨城県) 伊東臣明 B)ずるかまし(宮城県) 坂井一弘(東京都) 千葉広道 19A)オリジナルコーヒーカップ(北海道) 飯田伸一(愛知県) 五月女優(広島県) 田村和廣 B)ツインビー(茨城県) 内田好則(京都府) 上野政幸 20バトルチェス(三重県) 水谷泰三 21A)Zero(愛媛県) 武智和彦(鹿児島県) 本真光 B)Misty3(茨城県) 地引秀和 原田大輔 22セレクトッドソーサリアン 1(長野県) 塚本隆司(岡山県) 横山博道(福岡県) 浜地啓 2(東京都) 松村一郎(神奈川県) 三沢弘之(山梨県) 深沢享広 3(茨城県) 程田勝也(兵庫県) 村上貴之(大阪府) 中山良樹 23ウインドブレーカー(北海道) 渋谷康則(東京都) 八木貴弘(神奈川県) 久崎圭(岐阜県) 山口忠(大阪府) 鈴木哲也 24「この木なんの木」のCD(茨城県) 染谷祐一(福岡県) 徳久雅人(大分県) 山田博

以上の方が当選されました。おめでとうございます。商品は順次発送いたしますが、入荷状況などにより遅れる場合もあります。また、公正取引委員会の告示により、このプレゼントに当選された方は、この号の他の懸賞には当選できない場合がありますのでご了承ください。

(価格はすべて消費税別です)

ポケコンでCARPGを

Matsui Shin
松井 信

おっと、100号記念にちなんでポケコンの記事も復活かな？
でも何をやるかという、実はテーブルトークのRPGを楽しむのに利用しちゃおうというお話なんです。使用するのは圧倒的シェアを誇るPC-E500シリーズです。お楽しみに。



CARPGとは、Computer Aided Role-playing-game、つまり、コンピュータを利用したRPGのことです。私がいま名づけました。コンピュータRPG(以下CRPG)ではありません。あくまでもテーブルトークRPG(以下テーブルトーク)のサポートを目的としています。

テーブルトークRPGとはなにか？

テーブルトークとは、机の上で多人数でやるコンピュータを使わないRPGです。というよりは、CRPGのほうをコンピュータ上でやるテーブルトークの真似ごとといったほうが正確です。

D&D(Dungeons&Dragons)などのテーブルトークは、最近になってようやくやっている人も増えてきたようですが、それでも実際にやったことのある人はまだ少ないようで、RPGといえばCRPGのようなゲームと思っている人も多いようです。しかし、CRPGはテーブルトークから戦闘システム部分とストーリー進行を抜きだしたもので、それはテーブルトークの楽しみのごく一部に過ぎません。

テーブルトークの楽しさとは基本的にロールプレイ、すなわち「ゴッソ遊び」の楽しさです。つまりRPGというからには、キャラクターを演じられることが必要です。

CRPGでは、キャラクターを動かしていこそすれ、演じているとはとうていいえません。ドラクエをしていて自分が(本当に)勇者だと思いながらやっている人はたぶんいないでしょう。

しかし、テーブルトークでは、あなたはガラスの仮面のごとく、完全にキャラクターになりきって、現実世界のようにファンタジーワールドの中を冒険することができるようになるのです。いくつかの作業と若干の想像力を必要としますが、こういったリアリティと面白さはCRPGの比ではありません。

テーブルトークの実際

とはいえ、テーブルトークにも問題点があります。ひとつは、1人ではできないという点、しかも、そのうちの1人は「マスター」と呼ばれる進行役にならなければいけません。そして、ある程度の時間(数時間以上)と、場所(人数+機のスペース)が必要です。そういえば、マニュアルとそのほか道具も必要です。

テーブルトークはCRPGのように買ってきてすぐにできるものではありません。

とにかく、マスターになる人が、シナリオと呼ばれる台本(のようなもの。ゲームの設定およびストーリーなどを書いたもの)によって、ゲームを進行し、その架空世界のすべての出来事を管理し、同時にプレイヤーの不条理な要求に対処するわけです。当然、かなりの負担がかかるので経験者が望ましいわけです。

一方、1人ひとりのプレイヤーは、「キャラクター」というゲーム上での仮人格、つまり、その世界での自分を持ちます。それには、強さ、魔法、持ち物、その他さまざまな属性が決められていて、その世界におけるキャラクターの個性を表し、その行動に一定の制限を与えます。この辺はCRPGと一緒にですが、CRPGでは戦闘に関係ない属性はほとんどないのに対して、テーブルトークには戦闘以外にもさまざまな属性が存在します。キャラクターというのはひとつの人格なのだから、これは当然でしょう。

以上、テーブルトークのいい点として、

- 1) 別人格を演じることができる
- 2) 実際にはない世界で遊ぶことができる
- 3) 破壊衝動(?)を満足でき、ミッションに成功したときはカタルシスが得られるということがあげられます。また、
- 4) 議論や会話の訓練になる
- 5) 多人数でわいわい遊べる
- 6) マスターになって、いいシナリオがで

きたときは自己顕示欲(?)を満足できるなどのメリットも忘れることができません。

これだけの利点を持つテーブルトークが、ボードゲーム界に与えた影響は大きく、SLGなどは駆逐されかかって、SLGの雑誌であったタクテクスなどは、本家が季刊になって、月刊のRPG雑誌を出しているほどです。

CARPGとは

前に述べたように、やはりマスターは大変です(同時にやりがいもあるが)。そうしたある日、疲れたマスターである私は、ひたすら作業をしていて思いました。

テーブルトークの問題点である「作業」は、多くは数値の処理という機械的な作業です。これをコンピュータ化してしまえば、マスターの負担は軽減し、本来のロールプレイに専念できるようになるんじゃないか。これが、CARPGなのです。

テーブルトークにおける作業は、次のように分類されます。

- 1) キャラクターを作る
- 2) シナリオを作る
- 3) ゲームをする

まず、1)ですが、この辺は作業というよりは楽しみに属するものなので、ワープロの利用ぐらいにとどめておきます。

次に2)ですが、シナリオを作るというのは、小説のあらすじを作るようなものです。

まあ、仲間内でやるんだったらストーリーはどこからパクってくればいいのですが、敵の設定、地図作成、ストーリーの記述といったところだけでもかなりの作業となります。

これは、市販のシナリオを買ってくればすむ問題ですが、何千円もする高いものだし、そんなにたくさん出ていません。それに、自作シナリオを成功させることこそがマスターの醍醐味だし。というわけで、この辺のCARPG化はそのうち取り上げた

いと思います。

そして、なんとといってもマスターがいちばん大変なのは、3)の実際のゲーム中でしよう（と私は思う）。

なにしろプレイヤーは何人もいるのにマスターは1人なのだから。戦闘場面でたくさんの敵キャラクタを操りながら、プレイヤーの受け答えをするのは、やっぱり大変なことです。たとえば、

プレイヤーA: ゴブリン6に3ダメージ!

マスター: はい。

プレイヤーB: オーガ3に12ダメージ!

マスター: はいよ。

プレイヤーC: 魔法かけるよお。ホールドパーソン! ゴブリン4と5!

マスター: はい（コロ、サイコロを振る）。5は止まった。それから?

プレイヤーA: そっちの番だよ。

マスター: そうか。じゃいくよ。ゴブリン1が、えーと誰の前? あ、そう。アーマークラスいくつ? (コロ) 当たった。えっとダメージは(コロ) 2ね。じゃ、ゴブリン2は……。

これをえんえんと繰り返すのだから、慣れれば機械的にできるとはいえやっぱり面倒くさい。ましてや徹夜でやっていたりすると、うっかりするとパニックになりかねません。

そこで、戦闘中の敵モンスターのヒットポイントや攻撃を、コンピュータに管理させようというわけです。このプログラムを次回掲載する予定です。

コンピュータはなにを使う?

ところで、CARPGに使うコンピュータはなにがいいか。それは実はポケットコンピュータなのです。

まず、学校なんかでやる時は持ち運びができなくてはなりません。その点、ポケコンなら持ち運びもできるし、値段も安く、また、高級電卓として使えるので無駄な投資にはなりません。それに、工学系の大学生のほとんどはポケコンを持っているでしょう。

こういうと、ポケコンなんて、という人もいるかもしれませんが、今のポケコンをなめてはいけません。シャープのPC-E500（または、PC-1480U、PC-1490U）は、X1のBASICのような（というよりもN88-BASICのような）強力なBASIC、パソコンにも引けをとらない高速性、40×4行の広い画面、32KバイトのRAMは一部をRAMディスクとして使用でき、RS-232C

ケーブルでパソコンにつなげる、などとてもなく強力なマシンなのです。

では、自宅でやるならパソコンでいいやという意見もあるでしょうが、テーブルトークではマスターの情報はプレイヤーに見せてはいけないことになっています。したがって、机の上にマスターに向けてディスプレイが載ることになり、普通の家ではちょっと苦しいでしょう。そのため、ポケコンのほうが都合いいのです。

テーブルトークを始めるには

現在、たくさんの種類のテーブルトークが市販されていますが、やはりおすすめはD&Dおよび、AD&D(Advanced D&D)です。したがってこの連載も、対象は基本的にD&D、AD&Dとします。

D&Dはやはり日本ではもっともメジャーで、サプリメント（追加シナリオ、その他ゲーム補助用のツール）が多く、またルールがシンプルなため初心者でもやりやすいという特徴があります。

しかし、実は米英ではAD&Dのほうが遙かにメジャーで、そのサプリメントの量はD&Dの比ではありません。ルールもD&Dより体系化され、より面白くなっています。なにぶん英語というハンデがありますが、高校生でも読める程度のものでからそれほど心配することはありません。日本語版も7月から出版されるはずですが、最初は誤植が多いと予想されるので、いっそのこと英語版を買っても無駄にはならないでしょう。

というわけで、ようやくテーブルトークを始めるわけですが、なにもしたことがない人がいきなりマスターを始めるのは大変です。しかし、誰かがマスターをやらなけ

ればいけません。しかし、なにかとんでもない間違いをする可能性もあります。

そのため、まず最初は（少なくともマスターをやる人は）どこかでテーブルトークを体験してくることをおすすめします。たとえば、どこでも高校、大学なら誰かしらはテーブルトークをしているものですから、友達のつてから仲間に入ってみるというのがひとつの手です。

ほかに、テーブルトーク関係の雑誌には地域的なテーブルトークサークルのメンバー募集が出てますからそこに連絡を取ってみるという手もあります。

マスターへの道

とにかく、マスターになる人はロールプレイとはなにかということを理解してはいけません。ルールを読むことも忘れずに。

それから、特にファンタジー系テーブルトークの場合、たくさんのファンタジー小説と、ヨーロッパの歴史書、そしてそれ以外にもたくさん小説も読んで素養をつけておきましょう。マスターというのは、作家にして脚本家、監督にして俳優というとてもやりがいのある総合プロデューサーなのです。

それでは来月はプログラムに入ります。

〈参考文献〉

D & Dがよくわかる本 富士見文庫 490円

D&Dが具体的にどのような手順で進められるのかわかる。D&D初心者にはおすすめ。

眠れる龍 現代教養文庫 720円

アメリカのゲーマーの生活がわかってなんとなくほのぼのする。ファンタジー小説としてもいい出来。

ドリームパーク 創元推理文庫(SF) 580円

マスターの内輪うけと評されるだけあり、マスターをやっている人には面白い。

ゲーム紹介(1)

Dungeons & Dragons

RTS.inc. (日本語版:新和)

テーブルトークといえばD&Dというぐらいメジャーなゲームで、特に日本ではほぼ主流となっている。とにかくルールが簡単で覚えやすく、初心者でもとりあえず20面サイコロを振って殴っているだけで十分楽しい。

しかしながら、古いゲームであるということとは否定できず、攻撃は最大の防御でありキャラクタのレベルが2桁になるころからなにか間違ったゲームへと発散していく傾向が多々ある。最高レベルである36のあとには、みんなで神様をやろうというルールまであるが、きっとただの冗談だろう。

通称、赤Dといわれるベーシックと、青Dといわれるエキスパートの2つの箱が最低限必要。



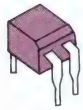
上級セットとしてコンパニオン、マスター、インモータルの拡張ルールセットがある。

基本インタフェース回路 その2

Misawa Kazuhiko
三沢 和彦

今回は製作実習編です。とても簡単な回路ですし、実体配線図も用意しました。注意事項も徹底的に詳しく解説してありますから皆さんも部品を揃えて実際に挑戦してください。うまくいったときの喜びは格別ですよ。

いよいよお待ちかねの製作実習編です。今月が待ち切れなくて、もう部品を揃えてしまった人もいいかもしれませんね。とにかく、まずは部品表のとおり部品を揃えてください。



汎用ケーブルの製作

最初にジョイスティックポートと自作回路をつなぐためのケーブルを作ります。このケーブルは1本作れば、連載で製作する回路すべてに使えるようにしてあります。圧着用の10ピンフラットケーブルとコネクタとは買ったお店で圧着してもらっておきます。部品を買うときに頼めば、その場で

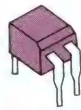
圧着してくれるはずですよ。

圧着されたコネクタを見ると、一番端に印がついているでしょう。これが1番ピンです。さて、このフラットケーブルを9ピンDサブコネクタにハンダ付けしていきます。9ピンDサブコネクタはメスコネクタでなければ、X68000につなげないので注意してください。

そして、Dサブコネクタの表に出る側をよく見ると、小さく1～9の数字が記されているのがわかるでしょう。そこで、圧着コネクタの1番ピンにつながっている線から順番にDサブコネクタの各端子にハンダ付けしていくのです。Dサブコネクタの端子どうしの間隔が意外と狭いので、ハンダ

が隣とくっつかないように注意してください。

ところで、ケーブルは10ピンでDサブコネクタは9ピンですから1本余ることになりますが、10番の線は9番ピンのGNDにいっしょにつないでおきます。ハンダ付けが無事終わったら10本のケーブルを束ねてDサブコネクタケースについている金具で止め、コネクタ全体をケースに納めます。これで出来上がり。



基本I/O基板の製作

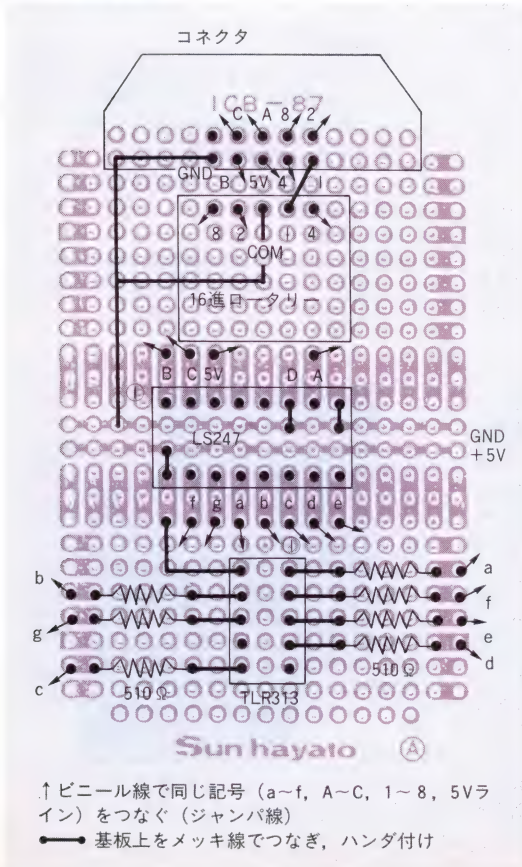
基板上に回路を組むときにもっとも頭を悩ませるのが、部品の配置です。部品の配置をうまく決めるかどうかで配線の手間がまったく違います。皆さんは図1の実体配線図を参考にしながら、以下の説明を読んでください。

サンハヤトのICB-87という基板はIC1個用の汎用基板で、ICの足まわりの配線がしやすいように工夫されているものです。次回に製作するA/Dコンバータもこの基板上に作るのです。何枚かまとめて買っておくのもいいでしょう。

●主な部品の取り付け

まず最初に、先ほど作った汎用ケーブルをつなぐ基板用コネクタを取り付けます。まずは10ピン全部をハンダ付けしてしまいます。このとき、ハンダ付け面から見て、ジョイスティックコネクタのピン番号は図2のように対応しています。ハンダ付けしたピンから各場所への配線はまだ行いま

図1 実体配線図



部品表

9ピンDサブメスコネクタ	1個	200円
Dサブコネクタケース (DE-C1-J6)	1個	360円
10ピンフラットケーブル	1m	100円
10ピンコネクタ (PS-SRN10)	1個	200円
IC用基板 (サンハヤトICB-87)	1枚	90円
10ピン基板用コネクタ (HIF3BA10P-DS)	1個	100円
16進ロータリースイッチ (アルプスSRRQ)	1個	250円
ICソケット16ピン	1個	35円
74LS247	1個	80円
TLR313	1個	210円
抵抗510Ω	6本	10円
ビニール配線材	少々	

図2 基板用コネクタ(ハンダ付け面から見た図)

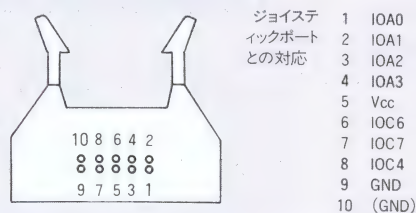
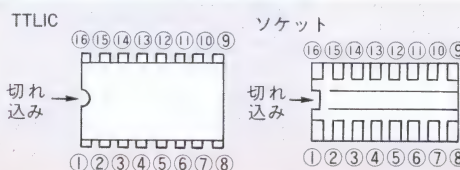


図3 IC、ソケットを上から見た図



せん。

次に、ICソケットを取り付けますが、ソケットを上からよく見ると図3のように片方に切れ込みがあり、これを目印にピン番号が決まっています。

規格表などに載っているICのピン番号はICの上から見たときのものなので注意してください。当然、配線している側から見ると逆回りになっています。この点は熟練者でも意外と勘違いすることがありますので油断しないように。もちろんこの連載では、実体配線図に従えばOKです。

次に、ICソケットを基板に差し込んだら、すぐに8番ピンと16番ピンを内側に折り込んでハンダ付けしてしまいます。というのも16ピンICの場合は8番がGND、16番が5Vに接続するのが一般的だからです。そして、基板がICB-87の場合は実体配線図を見てもわかるとおりICの2列の足の間にGNDラインと5Vラインの2本の配線ラインが通っているの、折り込んだ8番ピンと16番ピンとをそれぞれそのラインにもハンダ付けします。

このようにIC工作では、GNDラインと5Vラインを先に通してしまうのが基本なのです。これができればあとはICの足1本1本をすべてハンダ付けしていきます。今回は6番ピンもGNDに落とすので、内側に折り込んでGNDラインにハンダ付けします。

次に7セグメントLED (TLR313) をハンダ付けします。これはソケットがないので直接基板にハンダ付けしてしまうしかありません。TLR313のピン番号は先月号にも載せてありますが、やはりハンダ付けする側から見ると逆回りになっていることに注意しましょう。

TLR313の10番ピンは5Vラインに直結ですが、1～4, 6, 8, 9番ピンは510Ωの抵抗を介してLS247につながるの、次に抵抗の配線を行うのが効率的です。配線の都合上、抵抗は実体配線図のように寝かして差し込み、TLR313側の足は折り曲げて、図4のようにTLR313の各端子まで伸ばしてハンダ付けしてやります。反対側は基板の端に並んでいる端子にハンダ付けしてやり、余った長さはすっぱり切り落としてしましましょう。

こうして7本の抵抗を付け終えたら、16進ロータリースイッチを取り付けます。私の手に入れたアルプス製のものは取り付け用の足も端子も位置としてはIC用基板に適したものですが、ただひとつ取り付け足が大きすぎて、基板の穴にはそのままでは入りません。そこで、錐(きり)を使って

取り付け位置の穴を少し大きくしてからはめ込みます。はめ込んだら端子をハンダ付けしてしましましょう。

ここまでくると、部品はすべて取り付けられたことになります。ここでセンスの鋭い人はお気づきでしょうが、工作では、配線の前にすべての部品を取り付けてしまうのが鉄則です。それは何度もうのように、部品の配置とバランスが工作の手間を決めているといえるからです。

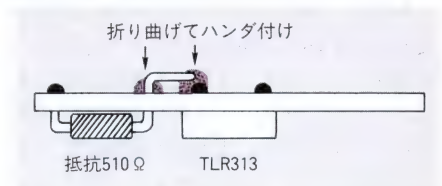
●部品間の配線作業

部品がすべて配置されたら、次は地道に配線作業です。まずは抵抗7本とICとの接続をしましょう。TLR313のa～gから伸びている各抵抗の端をLS247のa～gに対応させて、被覆されたビニール線でつないでいきます。このようにつなぎたいところどうしをジャンプしてつないでいる線のことをジャンプ線といいます。

このとき、TLR313のa～gの並び方も間違えやすいですし、そのうえ、LS247のa～gが9～15番ピンに割り当てられていますが、これも順番に並んでいないので注意が必要です。実は、私も最初は間違えてつないでしまいました。間違えてつなぐとLEDの表示がおかしくなりますが、壊れることはありません。それから、10番ピンを5Vラインにつなぐのも忘れないように。以上で、LEDまわりの配線は完了です。

次にLS247の入力1, 2, 7番ピンの配線です。ここは、10ピンコネクタに直結しますが、10ピンコネクタのピン番号も間違いないので、再度図2を確認してくださ

図4 TLR313と抵抗のハンダ付け



い。このピンは位置も連載のすべての回路に共通です。

ところで、今回の製作でいちばん難しいのがこの10ピンコネクタまわりの配線でしょう。隣と近いうえ、ビニール線がかさばるので、次に述べるように手際よく行います。まず、コネクタ側の端子はあらかじめハンダ付けしておくこと。そして、ビニール線の被覆を必要な分(1mmほどで十分)だけワイヤストリッパでむいておき、そこにもハンダを付けておきます。

このように、ハンダ付けする両側にあらかじめハンダを付けておくのがコツです。あとは、ハンダゴテを基板側に当て、ビニール線の先をハンダ付けしたい箇所につけるだけで意外とうまくできます。万一隣にもくっついてしまった場合には、ハンダ吸い取り器で完全にハンダを取り除き、最初からやり直します。一度失敗したハンダは、二度とくっつかないことを肝に命じておく必要があります。

最後に3番ピンを5VラインにつないでICまわりの配線は終わりです。4, 5番ピンはなにもしないでおきます。参考までに3～5番ピンの機能を囲み記事の中に記しておきますので、なにか自分で設計工作す

抵抗のカラーコード

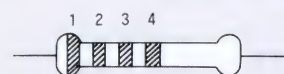
抵抗1本1本をよく見ると色のついた4本の帯が見えますが、これが抵抗値を示すカラーコードです。精度の高い特別な抵抗には5本ついているものもありますが、ここでは一般によくみかける4本組の読み方を説明します。

最初の3本が抵抗値そのものを示し、最後の1本は抵抗の精度を示しています。精度というのは、表示されている値を基準にして実際の抵抗値にどれだけ誤差があるかということです。たとえばそれが金色のカラーコードであれば、実際の抵抗値は表示値の±5%という意味ですから、100Ωの抵抗の場合なら、実際は95～105Ωになっています。

最初の3本の見方をマスターしましょう。そこは0～9の9種類の色で表されていて、1本目と2本目とで2桁の値を示し、3本目でさらに10の何乗倍かを示します。図中の例題で確認してください。皆さんは、0～9が何色に対応するか覚えましょう。それには、0から9までならべて「くちあだき、みあむはし」と語呂で覚えます。それぞれの色名の頭文字を並べただけですが、なかなか覚えやすいと思います。

	1本目	2本目	3本目	4本目
黒(く)	0	0	10の0乗 = 1	
茶(ち)	1	1	1 = 10	±1%
赤(あ)	2	2	2 = 100	±2%
橙(だ)	3	3	3 = 1,000	
黄(き)	4	4	4 = 10,000	
緑(み)	5	5	5 = 100,000	
青(あ)	6	6	6 = 1,000,000	
紫(む)	7	7	7 = 10,000,000	
灰(は)	8	8	8 = 100,000,000	
白(し)	9	9	9 = 1,000,000,000	
金			10の-1乗 = 0.1	±5%
銀			-2 = 0.01	±10%

カラーコードの位置



	1	2	3	4	
例1	茶	黒	赤	金	$1 \times 10^2 = 100 \Omega (1k \Omega)$
例2	緑	茶	茶	金	$5 \times 10^1 = 510 \Omega$ (今回使っているもの)

ときの参考にしてください。

あとは、10ピンコネクタの残りの端子を処理してやれば、完成です。そこでまず、16進ロータリースイッチとの配線を行います。ロータリースイッチには端子が5本出ているはずですが、1本は共通端子で、これはGNDラインにつなぎます。残りの4ビット端子はどの順に最下位ビットから並んでいるかあらかじめチェックしておいてから、10ピンコネクタの1～4番端子につなぎます。

店で品物を買うときに各端子の機能を尋ねておくのが得策です。自分で調べることになってしまったら、まずロータリースイッチを1に合わせておいて5本のうちの2本が導通しているかをテスターで計り、次に2、4、8と順次合わせて、やはりどの2本が導通しているか調べます。1、2、4、8すべての場合に共通な端子がGNDにつながり、あとはそれぞれのビットに対応するかチェックします。

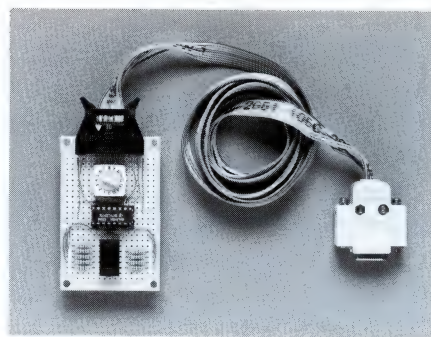
こうして、下位ビットから順に1～4ピンにつなげばOKです。実体配線図では、下位ビットから順に1、2、4、8、COMと記号を打ってありますが、品物によって

位置が変わるかもしれません。最後に、10ピンコネクタの5番と5Vライン、9番とGNDラインとをつなぎます。この5VとGNDとを逆にするとICが死ぬこともあり得ますから、気をつけてください。

●完成後のチェック

以上ですべての配線が終了し、いよいよ完成です。配線が終わったら、実際にX68000につなぐ前にもう一度実体配線図と比べて、配線のチェックをしてください。ただし、一度配線が終わってからチェックするまでの間にはお茶を飲むなりゲームをするなり、なにか気分転換をすることが大切です。最初は必ずどこか配線ミスをしているものですが、これを発見するためには頭を冷やしたあとのほうがずっと効率が良いのです。

十分チェックしたら汎用ケーブルと基板のコネクタをつなぎます。コネクタには片側に出っ張りがあり、これで上下の向きが決まっていますので、向きに注意しながらしっかり差し込んで最後にフックで挟み込んで止めます。そしていよいよX68000のジョイスティックポート1に差し込んでみましょう。



これで完成だ！

どうですか、LEDに3が表示されましたか？ もし3が表示されなければ、まだどこかにミスがあります。ただし、このテストは必ずX68000の起動直後にX-BASICを立ち上げて行ってください。

とりあえず、あり得るミスについて考えてみましょう。

1) なにも表示されない場合

5VラインとGNDラインの配線ミスです。単にどこかの配線し忘れか、もしくは5VとGNDとを逆にをつないでしまっているかもしれません。

2) LEDは点灯するが、表示がおかしい

TLR313のa～gとLS247のa～gとの対応がきちんとなっていない。あるいは隣どうしのピンがショートしていることもあり得ます。

3) 表示はするが、3でない場合

Dサブコネクタか基板の10ピンコネクタまわりの配線ミス。LEDまわりの配線はOKです。

以上、どうしても配線ミスが見つからなければ、ICが死んでいることも考えられますが、実際のところICの不良は万に一つしかないと思って差し支えありません。根気よくミスを探してください。

*

いかがでしたか？ まったく初めて工作する人でもこの程度の回路なら十分ついて行けるのではないのでしょうか。完成したらさっそくX68000からコントロールしてみたいところですが、はやる気持ちを抑えて次回までのお楽しみとしましょう。

来月はまず、ソフトウェアで最も基本となるI/OコントロールドライバをX-BASICの外部関数の形で提供します。といってもたいてい難しくないプログラムです。最初は68000アセンブラ入門みたいな解説になるでしょう。そのあとにそのドライバを使った応用プログラムを作ってみます。同時に一般的なI/Oコントロールを行うための基本もきっちり押さえる予定ですのでお楽しみに。ではまた、来月。

LS274の機能

図は規格表からの抜粋です。この図を見ながら各ピンの機能を順番に説明しましょう。

●電源系統

まず+5VとGNDは問題ないと思います。

●出力

出力（9～15番ピン）は7セグメントLEDのa～gに対応して、抵抗を介して接続します。図中に小さくa～gが書かれているのがわかるでしょうか。

●入力

入力は上位ビットからDCBAの順になっています。4ビット入力なので、0～15まで入力できますが、10以上になると意味のない表示になってしまいます。また、このICは表示の機能しかないのです。たとえば桁上りを自動的に足し込むようなことはできません。

●オプション

3番ピンはランプテストといって、ここをGNDに落とすと強制的にすべてのセグメントを点灯させます。通常は5Vラインにつないでおきます。

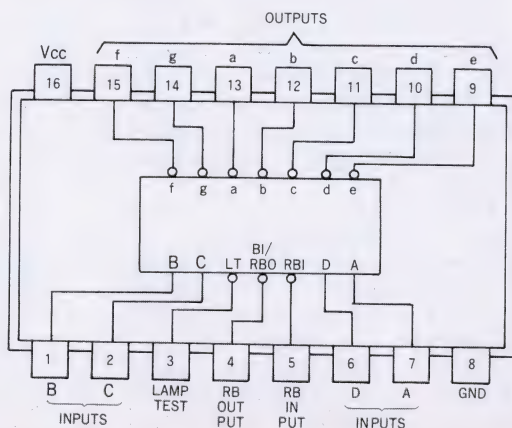
4番ピンのRBOと5番ピンのRBIはリップルブランキング機能に使うもので、通常はやはりなにもつないでおきます。これはたとえば、4桁のLEDに3桁の数字を表示させるとき、上の桁の0を表示させないようにするために使います。

それには、最上位桁のRBIをGNDにおとし、そこから順に上の桁のRBOを次の桁のRBIにつないでいきます。RBI

がLのときは、もし入力が0ならばなにも表示せず、しかもRBOからLを出します。

上の桁から順に0のときだけ数珠つなぎでなにも表示させないようにRBOをLにして伝達していきますが、ある桁で0でないときはそこから先はRBOがHのまま伝わっていくので途中の桁が0であっても0を表示します。

言葉で書くとなんとなくわかりづらいかもしれませんが、もし電卓や時計などで数桁にわたって表示させたいときにはそれぞれの用途で専用の表示用ICが簡単に手に入りますので、LS247のこの機能について理解しなくてもかまいません。それでも興味ある人は各自規格表を見て自由研究としてください。



超入門・ファイル処理

Izumi Daisuke 泉 大介

このところ難易度の高くなってきた調理実習ですが、今回は基本にかえて簡単なファイル処理の方法を解説しましょう。また、応用としてYETのスコアファイルを複数のプレイヤーで使用するためのアレンジも行っています。挑戦してください。

ゲーム作りもひと段落ついたところで、今月はちょっと実務っぽくファイル処理に取り組んでみたいと思います。一般にファイルというと、書類を綴じ込んだものを指しますが、コンピュータの世界ではディスクに保存されているBASICのプログラムやワープロの文書、表集計ソフトやデータベースのデータなどのことをいいます。文字やデータを綴じ込んだものだと見れば、なるほどファイルと呼ばれるのもうなづけるような気がします。この対比でいくと、ディスクドライブはさしずめファイルキャビネットというところでしょうか。

これらのファイルの内容に対するさまざまな作業がファイル処理で、簡単なところではファイルの中から単語を検索し、その単語が含まれている行だけを抜き出す。ファイルに入っている文字数、単語数、行数を数える、といった作業があります。ファイル処理というと難しそうな印象を持たれるかもしれませんが、コツをつかんでしまえば実に簡単なものなのです。なんせあのC言語では入門編で取り上げられる程度の題材なのですから。

ファイルのオープン、クローズ

ファイルを扱うときの儀式として、ファイルのオープン、クローズという作業があります。紙綴りファイルから必要な情報を探し出すときにはファイルを開きますね、また、作業が終わればファイルを閉じてキャビネットに戻します。これに対応するのがファイルのオープン/クローズです。ディスク上のファイルに開くも閉じるもないような気がしますが、コンピュータにとっては別の意味を持っています。

ファイルをオープンするとは、このファイルを使っているよとコンピュータに宣言する作業です¹⁾。これによってコンピュータはそのファイルが使用中であると認識し、ほかの人が同じファイルを使おうとするとエラーを出すことができるようになります。ファイルキャビネットなら使用中のファイルはキャビネット内にはありませんが、ディスクでは使っ

ていようがいまいが常にキャビネット内にファイルがあるようなものですからね。さらに進めて、見るだけなら何人の人が同時に見ようとディスク上のファイルが変更される心配はありませんから、複数の人がオープンできるようにすることもできます²⁾。

逆に、ファイルをクローズするというのは自分が使い終わったことをコンピュータに知らせるための作業です。X-BASICを終了するとオープンされたままのファイルは自動的にクローズされますが、自分でオープンしたファイルは必ず自分でクローズするようにしたいものです。

●見るのか、更新するのか、作るのか

ファイルを使うといってもいろいろあります。ファイルをオープンするときには、そのファイルを見るだけなのか書き込みをするのか、すでに存在するファイルを扱うのか新たに作るのかを明確にしなければなりません。これが「アクセスモード」あるいは単に「モード」と呼ばれるものです。

X-BASICでは「読む」「書く」「読み書きする」「新たに作る」という4つのモードでファイルをオープンすることができます。ワープロならば最初に文書ファイルを読むだけ読んで、変更が終わったあとに今度はすべて書き出せばOKですが、随時データの読み書きが行われるデータベースではファイルは「読み書き」モードでオープンする必要があります。メモリに入りきらないほどの大きなデータベースもあります。こうなると「読み書き」以外には扱う方法がありません。

●ファイル番号でファイルを管理

X-BASICでファイルをオープンするときには、`fopen` (ファイル名, モード) とします。モードは `r` (読む), `w` (書く), `rw` (読み書き), `c` (作る) と文字で指定するようになっています。たとえば、`myfile` というファイルを作りたいのなら、

```
fopen("myfile", "c")
```

となります。

ファイル処理を考えると、ファイルをひとつだけ

1) ファイル管理を一手に引き受けているのはOS(X68000ではHuman68k)ですので、ここは正確にはHuman68kに宣言する、となります。

2) マルチユーザーのOSではこの機能は必須といえるでしょう。X-BASICではファイルがすでにオープンされているかどうかのチェックすら行いませんが……。



しかオープンしないというのは稀です。あるファイルから特定の文字を探し出し、その文字列を含む行を別のファイルに書き出すというように、2つあるいは3つのファイルを同時にオープンして使うのが普通です。

ファイルがひとつだけならデータを読み込む、書き出す対象がどのファイルなのか迷うことはありません。しかし、オープンされているファイルが複数になると、対象がどのファイルなのか特定できなくなります。読み書きのたびにファイル名を指定するというのもひとつの解決法ですが、プログラムを書くのが面倒ですし、さらに1文字書くたびにファイル名の比較をやって対象のファイルを特定することになるので時間がかかってしかたありません。

そこでファイル番号³⁾の登場です。オープンしたファイルに番号を付けておいて、あとはこの番号を利用して読み書きを行おうというものです。X-BASICではファイル番号はファイルをオープンしたときにfopen関数の戻り値として返される整数です。次の命令を試してみてください。

```
print fopen("test","c")
```

これでtestというファイルが新たに作成され、返されたファイル番号が画面に表示されるはずです。ファイル作成を指定すると、すでに存在するtestというファイルを消去して新たに作ってしまいますので注意してください。実際にはこのファイル番号を変数に入れておきます。

```
int file
```

と宣言し、表示された値を代入しておきましょう。

続けてもうひとつファイルを作ってみます。

```
int file2
```

```
file2=fopen("test2","c")
```

変数file2を表示して、返されたファイル番号を確認してみましょう。

オープンしたあとのファイル操作はすべてファイル番号を使うと説明しました。ファイルのクローズも例外ではありません。クローズにはfclose関数を使い、引数にファイル番号を指定します。

```
fclose(file2)
```

なら、test2がクローズされます。もちろんtestはまだオープンされたままです。fclose関数はファイルを個々にクローズするのに便利な関数です。クローズ用の関数にはもうひとつfcloseallがあります。これはオープンされているファイルをすべてクローズするので楽ちんです。では、次に進む前にfcloseall関数でファイルを全部閉じておくことにします。

```
fcloseall()
```

と入力すれば現在オープンされているファイルtestも（もしmyfileをオープンしているならそれも）クローズされます。

データを読み込んでみよう

さあ、いまや皆さんはファイルを開けたり閉じたりする方法を修得したわけです。fopen、fcloseという2つの関数はファイルの世界に入る最も基本的な呪文です。覚えた呪文はすぐに使って慣れるのがマジックポイント向上の秘訣とばかりに、さっそくfilesコマンドで表示されるファイルを片っ端から読み出しモードでオープンしている方もいらっしゃるでしょう。

そんな向上心旺盛なあなたに質問です。オープンできないファイルはありましたか？ X-BASICの世界、すなわちHuman68kの世界にはこの方法でオープンできないファイルは存在しません。どんなファイルでも（それがfilesコマンドで表示されるファイルなら）オープンすることができるのです。ワープロの文書ファイルやBASICで作ったプログラムのファイルはもとより、皆さんが使っているX-BASICもオープン可能です⁴⁾。ワープロの文書やBASICのプログラムファイルは文字の集まりです。これに対しX-BASICはマシン語で書かれたBASIC本体です。X1のHuBASICなどではこういったマシン語プログラムファイルはオープンすることができませんでしたが、X-BASIC（すなわちHuman68k）ではなんの制限もありません。文字が収められたファイルも実際にディスク上ではASCIIコードの集まりです。つまり1文字単位で読み込めば、0~255の数値が返ってくるだけなのです。ダンプリストでお馴染みのマシン語は16進数2桁（これも0

3) Human68kではファイルハンドルと呼んでいます。またファイルを示すものという意味でファイルポインタと呼ぶ場合も多々あります。ファイル番号を保持する変数にfpという名前が多いのはこのファイルポインタを略したものです。

4) X-BASICはBASIC.Xというファイル名でBASIC2ディレクトリ（あるいはBASICディレクトリ）に入っています。

～255の数値)の集まりですから、ファイル内では両者はまったく同じものだといえます。どんなファイルでもオープンできるというのはX-BASICのファイル処理の大きな特長です。

● 1 文字単位で読み込む

では実際にファイルからデータを読み込んでみることにしましょう。まずは適当なプログラムを作り、それをTEST.BASというファイル名でセーブしてください。以前作ったプログラムがある方はそれを使って結構です。

まずはファイルのオープンです。データを読み込むのですから“r”でオープンします。

```
int file
file=fopen("TEST.BAS","r")
ですね。
```

さて1文字単位の読み込みですが、これにはfgetcという関数を使います。cはcharacterを意味しています。先ほど触れたように、この関数は文字を返すのではなく、ASCIIコードを返してきます。

```
print fgetc(file)
```

を実行してみてください。先ほど適当に作ってセーブしたプログラムの最初の文字のASCIIコードが表示されます。行番号の前にはスペースが詰まっていますから、スペースのASCIIコード32 (20H) が画面に表示されたはず。このままではわかりづらいので、chr\$関数でASCIIコードを文字に変換することにしてしましましょう。これは、

```
print chr$(fgetc(file))
```

でOKですね。ファイルの最後まで続けて表示するのなら、

```
while 1:print chr$(fgetc(file));:
endwhile
```

となります。セーブしたプログラムが表示され始めましたね。プログラムの最後まで表示すると……

「ピッ! (エラー音)」
ハイ、エラーです。

エラーが発生してしまいました (たぶん「バイトの範囲を越えました」と表示されているはず)。表示されたエラーメッセージを見てもなにが起こったのかわからないでしょうから解説しましょう。これはファイルの最後まで到達したにもかかわらず、さらにデータを読み込もうとしたのが原因です。ファイルの最後まで達すると、fgetc関数は-1を返します。

```
print fgetc(file)
```

として試してみましよう。ところがchr\$関数はchar型の引数(0～255)しか受け付けません。つまりchr\$(-1)を実行したのと同じことになりエラーが出たのです。ファイルを最後まで読み込んだら、それ

れ以上読みに行かないようにプログラムする必要があります。

ファイルの最後に到達したかどうかを調べるにはfeofという関数を使います。この関数は、

```
feof (ファイル番号)
```

という書式で利用し、指定されたファイルが最後まで (end of fileまで) 達していたら-1を、まだ達していなかったら0を返します。これを使って、

```
while feof(file) <> -1 : ~: endwhile
```

と先のwhileループを書き直せば、ファイルの最後まで文字を表示し続けることができます。fcloseall関数でTEST.BASファイルをクローズし、もう一度ファイルのオープンからトライしてみましょう。今度はエラーも起こりませんね。最後に、

```
fclose(file)
```

でTEST.BASをクローズすれば、ファイル処理入門はめでたく終了です。

リスト1はマシン語ファイルを表示するためのプログラムです。マシン語ファイルはchr\$で変換しても意味のある文字にはなりませんから、ダンプリストにならって2桁の16進数で表示することにした。また数値がずらずらと並んでいるだけというのは見苦しいので、データ16個ごとに改行するようにしてあります。while～endwhileループでファイルエンドまで回しながら、for～nextを使って16個のデータを表示するという方法でプログラムしました。基本的には上の文字表示のプログラムと同じですからすぐにわかると思います。

このプログラムを使って、TEST.BASを表示してみましょう。2桁の16進数がずらずらと表示され、なにが入っているのかさっぱりわからないかもしれませんが、注意して見るとところどころに「0D 0A」というデータが入っているのがわかると思います。この2つのデータは改行を意味し、プログラムをロードするときX-BASICはこのデータを手掛かりに行の終わりを判定しているのです。

リスト1 マシン語ファイルを見る

```
10 str filename          /* ファイル名
20 int file, data         /* ファイル番号、データ
30 int readingFlag=1     /* 読み込み中フラグ
40 int i
50 /*
60 input "ファイル名: ",filename /* ファイル名入力
70 file=fopen( filename, "r" ) /* ファイルオープン
80 while readingFlag     /* 読み込み中は以下を実行
90   for i=1 to 16       /* 16回繰り返す
100    if feof( file ) = -1 then {
110      readingFlag = 0 /* ファイルエンドならフラグ
120      break          /* をクリアしてループ中断
130    }
140    data=fgetc( file ) /* データを1つ読み込み
150    print hexStr( data );" "; /* 16進で表示
160  next
165  print              /* 次の行へ
170 endwhile
180 fclose( file )      /* ファイルを閉じて
190 end                  /* 終了
200 /*
210 func str hexStr( data ) /* 16進2桁の文字にする
220   return( right$( "0"+hex$(data), 2 ) )
230 endfunc
```


システムディスクのBINディレクトリにはDUMP.Xというプログラムが入っています。これは

```
69 6E 70 75 74 20 22..... input " .....
```

というように、16進数とそれをASCIIコードと見なしたときの対応する文字を表示してくれます。リスト1はこの左半分だけを表示するようなものです。リスト1を改造し、DUMP.Xのような出力ができるように挑戦してみてください。

●データを読み込むそのほかの関数たち

X-BASICではfgetcのほか、freads、freadの2つの関数でデータをファイルから読み込むことができます。freadsは文字が入っているファイルを対象とし、改行コードまでの1行を一気に文字変数に読み込む関数です。1文字1文字読み込むより一気に読むほうが速いので、文字ファイル処理では多用される関数です。もう一方のfreadは1次元の数値型配列を一気にファイルから読み込む関数です。

●データを書き出す

ファイルにデータを書き出すときには、“w”モードか“c”モードでファイルをオープンし、データ書き出し用の関数を使うだけで基本的な作業はまったく同じです。データ書き出し用に用意されている関数はfputc、fwrites、fwriteの3つで、これまでに紹介してきた読み込み用関数と対になっています。

fwriteは実験データなどを1次元の配列に収めておき、「ハイ、セーブ!」と一発で処理できる便利な関数です。

YET再び

6月号付録ディスクのYET.Xはトップ10のスコアをファイルに残します。もともとオマケ的な要素が強かったので暗号化も行わず、単純に名前とスコアを記録するようになっています。DUMP.Xで覗くとその構造がよくわかるでしょう。作成当時には最高得点は3万点台が限界だろうと思い、このあたりなら十分自分の名前を残すことができるという自負から、同じ人物の得点は最高点のみを残すなどという細工を行わなかったのです。

ああそれなのに、それなのに。編集室ではいつしかトップ10すべてが4万点台になってしまったのです。しかもたった2人の人物によって! 結局私はやってもやってもスコアを残すことができず、「これはなんとかしなければ」という使命感のもと、スコア調整プログラムを作ることにしました。

このスコア調整プログラムは次の2つの機能を持っています。

- 1) 同じ人物のスコアは最高得点のみを残す
- 2) 2つのスコアファイルを融合する

1)は1人の人物がスコアを独占し、ほかの人が名前を登録する榮譽にあずかれないという事態を打破するために用意しました。2)は自宅でさんざんやって出した高得点をクラブのX68000に移し、友達に尊敬されるためです。

●YETSCOのファイル構造

YETのスコアファイルであるYETSCOは整数型の配列をfwrite関数でファイルに書き出しただけの非常に簡単な構造をしています。1人分のデータは、

1. 名前の1文字目のASCIIコード
2. 名前の2文字目のASCIIコード
- :
6. 名前の6文字目のASCIIコード
7. スコア

という形式で7つの整数型データに変換され、これが10人分続いたのがスコアファイルなのです。例をお見せしましょう。「DAISKE 32000」というスコアをこの方法で変換すると、

```
68 65 73 83 75 69 32000
```

となります。

●まずはスコアファイルの読み込みから

ではまず、スコアファイルの読み込みです。スコアファイルはfwriteで書き出したファイルですので、読み込みはfreadで行いましょう。1人のデータが整数7個分ですから、10人のデータは整数70個分になります。

```
int scoFile(70)
```

でデータを読み込む1次元配列を作成し、

```
int file  
file=fopen("yetsco","r")  
fread(scoFile, 70, file)  
fclose(file)
```

でデータの読み込みは終了です。freadは読み込む配列名と、読み込むデータの個数、そしてファイル番号を引数にとります。

●名前とスコアを取り出す

データをいったん読み込んでしまえば、あとは普段のプログラミングと変わりありません。これまで初期値を与えた配列を使うプログラムをいくつか作ってきましたが、初期値を与える代わりにファイルから読み込んだだけだと考えてもいいでしょう。

いま、(ファイルから読み込んで)初期値を与えた配列scoFileがあります。これはASCIIコードとスコアをごちゃまぜにして登録してある配列です。このままでは扱いづらいので、プレイヤーの名前を入れた配列と、スコアを入れた配列に分けることにします。

```
str player(9)  
int score(9)
```


の2つの配列を用意し、これにscoFileのデータを取り出してセットします。

プレイヤーの名前は必ず6文字分としてscoFileに収めてあり、そのあとにスコアがセットしてありますから、

```
for i=0 to 9
  for j=0 to 5
    player(i)=player(i)
    +chr$(scoFile(i*7+j))
  next
  score(i)=scoFile(i*7+6)
next
```

として2重ループを作ればplayerとscoreの2つの配列にデータをセットすることができます。6つの名前データと1つのスコアデータが1組になっていますから、i*7番目から6つのデータを取り出しそれを文字列に変更してplayer配列のi番目に、その次のデータを取り出してscore配列のi番目にセットしているのが上のプログラムです。

ではここでプログラムを見ていただきましょう。リスト2です。10行ではスコアファイル名をユーザが設定できるように変数として宣言しています。20行はいま説明したデータ読み込み用配列、そして30、40行がplayer配列とscore配列です。ここでは3つ宣言してありますね。これは、このあと同じ名前の削除を行うのに、ひとつの配列の中でやりくりするのは面倒なためです。加工後のデータは別の配列に入れることにしました。

上で説明したファイル読み込みおよびplayer, score配列へのセットを行っているのは1160行のreadSco関数です。ここでは引数nの値によって、player1, score1にセットするのか、player2, score2にセットするのかを振り分けています。

●同一人物の削除

同一人物を削除するには、同じ名前を飛ばしてスコア配列を詰めていけばOKです。readSco(1)でplayer1, score1配列にデータを読み込み、player1配列を上から順に見ていって、初めて登場する名前なら名前とスコアをplayer, score配列へ移します。

問題は初めて登場する名前かどうかを判定する方法です。player配列を順に調べてもいいのですが、ここではinstr関数を使うことにしました。instr関数は、文字列が特定の文字列を含んでいるかどうかを判定する関数です。player1配列からplayer配列へ移した名前を文字型変数chkStrに順次代入していくことにすれば、ある名前をplayer配列に移したかどうかはchkStrを調べるだけですみます。

ここで気をつけなければならないのは、文字列を単純に追加してはいけないということです。スコア

のトップがdai, 2番目がdanだったとします。単純に追加するとchkStrは、

daidan

となりますね。スコアの3番目がidaだとすると、idaはすでにchkStrに入っていることになってしまうため、player配列へ移されません。

このような事態を避けるため、名前の前後を決して名前に使われない文字で区切る必要があります。決して名前に使われない文字を仮に‘.’だとすると、chkStrは、

.dai.dan.

となり、“.ida.”はこの中に含まれないのでうまくいきます。

これらの処理を行っているのが320行から始まるunify関数です。ここでは区切り文字としてchr\$(1)を使っています。

●整頓後のスコアの保存

スコアの保存はスコアの読み込みと逆の手順で行います。整頓が終わったスコアはplayer, score配列に収められていますから、これら2つの配列からscoFile配列へデータを移し、それをfwrite関数で一気書き出せばOKです。これは1570行のsaveSco関数が行っています。

●2つのスコアファイルを融合する

player2, score2配列が用意してあるのは、この機能を実現するためです。1つ目のファイルをplayer1, score1配列に、2つ目のファイルをplayer2, score2配列にセットし、これら2つの配列から点数の大きいもの順にplayer, score配列へと移していくと融合が完成します。具体的には2つの配列の添字用に2つの変数(rank1, rank2)を用意し、

```
if score1(rank1)>=score2(rank2) then {
  配列 player1, score1をplayer, scoreへ
  rank1=rank1+1
} else {
  配列 player2, score2をplayer, scoreへ
  rank2=rank2+1
}
```

とします。添字変数はデータを移したときだけ大きくなり、次のスコアがもう一方のスコアと比較されることになります。

この処理を行っているのが730行から始まるmerge関数です。画面表示処理が間に入っているのですが、若干わかりづらいかもしれませんが、やっていることは上で説明したことだけです。

●プログラムの拡張について

さて毎度のことながら、プログラムには必要最小限の機能しか盛り込んでありません。エラー処理はまったくやっていませんし(X-BASICで実行する

なら、致命的なエラーはBASICが出してくれる), 処理を途中でやめたくなった場合のことも考慮してありません。整頓終了後に画面に表示される結果が気に入らない場合は、ファイル名入力のプロンプトが表示されているときにブレイクしてください。

まず最初に皆さんに取り組んでもらいたい拡張は、2つのファイルを融合するときに同一人物を削除する機能を付加することです。unify関数が参考になるかと思います。

yetscoファイルを読み込み変数にセットすることのプログラムを使えば、簡単にスコアを変更することができてしまいます。player, score配列の中身を適当にいじってsaveSco関数を呼び出すだけでいいのですから5万, 10万点のスコアなんて楽勝です。そんなスコアを見せびらかして喜ぶような悲しい遊びはやらないでくださいね。

来月は「ちょっと高度なファイル処理」と称してデータベースもどきをお送りする予定です。

リスト2 YETのスコア管理ぶろぐらむ

```

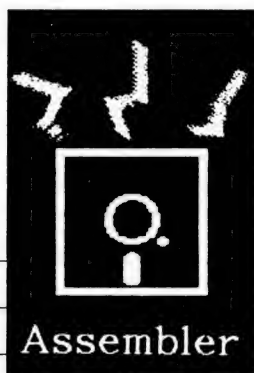
10 str scoName          /* スコアファイル名用
20 int scoFile(70)      /* データ読み込み用1次元配列
30 str player(9), player1(9), player2(9)
40 int score(9), score1(9), score2(9)
50 int conFlag = 1
60 str selection
70 /*
80 while conFlag
90   cls
100  print "Score Manager"
110  print
120  print "1) 同一人物削除"
130  print "2) ファイル融合"
140  print "3) 終了"
150  print
160  print "処理する番号:";
170  selection = inkeys
180  switch selection
190    case "1"
200    unify()
210    break
220    case "2"
230    merge()
240    break
250    case "3"
260    conFlag = 0
270    break
280  endswitch
290 endwhile
300 end
310 /*
320 func unify()
330   str chkStr[80]
340   int i, rank=0
350   /*
360   /* スコアファイル読み込み
370   /*
380   locate 0, 10
390   input "スコアファイル名:", scoName
400   readSco( 1 )
410   cls
420   for i=0 to 9
430     print player1(i), score1(i)
440   next
450   print
460   /*
470   /* 同一人物削除
480   /*
490   print " 変換"
500   print
510   chkStr = chr$(1)
520   for i=0 to 9
530     player(i) = ""
540     score(i) = 0
550     if instr( 1, chkStr, chr$(1)+player1(i)+chr$(1) ) = 0 then
560       chkStr = chkStr + player1(i) + chr$(1)
570       player(rank) = player1(i)
580       score(rank) = score1(i)
590       rank=rank+1
600   next
610   next
620   for i=0 to 9
630     print player(i), score(i)
640   next
650   print
660   /*
670   /* スコアファイル保存
680   /*
690   input "セーブします。ファイル名:", scoName
700   saveSco()
710 endfunc
720 /*
730 func merge()
740   int rank, rank1, rank2
750   /*
760   /* 2つのスコアファイル読み込み
770   /*
780   locate 0, 10
790   input "スコアファイル名1:", scoName
800   readSco( 1 )
810   input "スコアファイル名2:", scoName
820   readSco( 2 )
830   cls
840   for i=0 to 9
850     print player1(i), score1(i), player2(i), score2(i)
860   next
870   print
880   /*
890   /* 2つのスコアを1つにまとめる

```

```

900  /*
910  print " 変換"
920  print
930  rank1=0 : rank2=0
940  for rank=0 to 9
950    if score1( rank1 ) >= score2( rank2 ) then {
960      player( rank ) = player1( rank1 )
970      score( rank ) = score1( rank1 )
980      rank1 = rank1 + 1
990    } else {
1000     player( rank ) = player2( rank2 )
1010     score( rank ) = score2( rank2 )
1020     rank2 = rank2 + 1
1030   }
1040 next
1050 for i=0 to 9
1060   print player(i), score(i)
1070 next
1080 print
1090 /*
1100 /* スコアファイル保存
1110 /*
1120 input "セーブします。ファイル名:", scoName
1130 saveSco()
1140 endfunc
1150 /*
1160 func readSco( n )
1170   int file
1180   int i
1190   str ch
1200   /*
1210   /* 対象とするスコア配列にデータ読み込み
1220   /*
1230   file = fopen( scoName, "r" )
1240   fread( scoFile, 70, file )
1250   fclose( file )
1260   /*
1270   /* 対象とする名前配列をクリア
1280   /*
1290   for i=0 to 9
1300     if n = 1 then {
1310       player1(i) = ""
1320     } else {
1330       player2(i) = ""
1340     }
1350   next
1360   /*
1370   /* 名前配列に名前を
1380   /* スコア配列にスコアをセット
1390   /*
1400   for i=0 to 9
1410     for j=0 to 5
1420       ch = chr$( scoFile( i*7 + j ) )
1430       if n = 1 then {
1440         player1(i) = player1(i) + ch
1450       } else {
1460         player2(i) = player2(i) + ch
1470       }
1480     next
1490     if n = 1 then {
1500       score1(i) = scoFile( i*7 + 6 )
1510     } else {
1520       score2(i) = scoFile( i*7 + 6 )
1530     }
1540   next
1550 endfunc
1560 /*
1570 func saveSco()
1580   int file
1590   int i
1600   /*
1610   /* 名前配列から名前を
1620   /* スコア配列からスコアを取り出し
1630   /* scoFileにセット
1640   /*
1650   for i=0 to 9
1660     for j=0 to 5
1670       scoFile( i*7 + j ) = asc( mid$( player(i), j+1, 1 ) )
1680     next
1690     scoFile( i*7 + 6 ) = score( i )
1700   next
1710   /*
1720   /* scoFileを書き出す
1730   /*
1740   file = fopen( scoName, "c" )
1750   fwrite( scoFile, 70, file )
1760   fclose( file )
1770 endfunc

```

マウスwithグラフィック

Murata Toshiyuki 村田 敏幸

X68000用のプログラムを作成するというのなら、やはり、マウスも基本として押さえておきたいところ。マウス制御のためのさまざまな機能がIOCSとして用意されていますからこれを利用するのが正攻法です。簡単なお絵かきツールで実践してみましょう。

最初に、前回の記事中にポカがあったので訂正しておく。アドレスレジスタにaddqやsubqで小さな定数を加減算するときにワードサイズを指定したほうが速いという書いたが、大嘘なので忘れてほしい。実際には、ロングワードでもワードでも実行速度は変わらない。また、最後のASX.Sの中で使っているインクルードファイルが抜けていた。リスト0にそのFILES.Hを示す。ひと月休んで訂正が遅れたことと合わせてお詫びする。痛惜の念に堪えない、ぐらいのことはいうべきなのかもしれないが、この言葉はいつかともない大バグを出したときのためにとっておこうと思う。

*

さて、今回は地味ながらX68000らしいところでマウスを取り上げ、最後はこれにパラパラとグラフィックを振りかけてこちんまりとまとめてみたい。あくまでマウスがメインであり、グラフィックまわりについてはあまり詳しく触れないことをあらかじめ断っておく。

IOCSコールを使う

X68000ではROMにIOCS (Input/Output Control System)の形でさまざまな機能の制御ルーチンが用意されており、マウスもこのIOCSを呼び出すことによってほとんどX-BASICと同じ感覚で手軽に利用することができる。もうご存じだとは思いますが、一応、IOCSの概要と呼び出し手順を押さえておこう。

X68000のIOCSはテキスト画面への文字表示、キー入力に始まって、プリンタ出力、RS-232Cによる入出力、フロッピーディスク/ハードディスクの物理的な読み書き、マウスの制御、グラフィック描画、スプライト、AD PCM、カレンダー時計などの周辺LSIの制御にいたるまで、X68000の(ほとんど)すべての機能を網羅している。位置づけとしてはシステム中もっともハードに近い部分を担当しX68000上のプログラムを底辺からささえる低レベルI/Oルーチン集¹⁾であり、OSであるHuman68kもIOCSに乗った形で作られている。

これにより、ユーザープログラムがHuman68kに入出力を要求すると、Human68kは必要に応じてIOCSを呼び出し、最終的にIOCSがハードに働きかけて物理的な入出力を行う²⁾。結果は逆のルートを伝って返される。このHuman68kとIOCSとの上下関係(というか依存関係というか階層構造というか)は心に留めておいてもらいたい。

IOCSを呼び出す手順はいたって簡単で、d0,1にIOCSコール番号を入れてtrap #15という命令を実行するだけだ。パラメータがあるときはd1以下のデータレジスタやa1以下のアドレスレジスタ(a0はIOCSコール呼び出しには使われない)に入れて渡す。たとえば、IOCSコール番号20Hに割り当てられている1文字表示機能を使うときには、

```
move.w    #'A',d1
moveq.l   #$20,d0
trap      #15
```

IOCSコール番号21Hの文字列表示機能を使うのなら、

```
lea.l     mes,a1
moveq.l   #$21,d0
trap      #15
:
```

1) 実際にはハードがらみ以外にも、シフトJIS漢字コード↔JIS漢字コードの相互変換とか、ユーザーモードからスーパーバイザ空間にあるメモリを読み書きするといったユーティリティ的なものもIOCSには用意されている。

2) 論理的には、OSの低レベルI/Oはデバイスドライバが担当することになっているわけだが、現実にはHuman68kのデバイスドライバはさらに下位の存在であるIOCSを下請けに使っている場合が多い。

trap命令

trapは端的にいうと故意に例外を発生させる命令だ。trapにはtrap #0~#15の16個があり、順に例外ベクタ番号20H~2FH、例外ベクタアドレスでいうと0080H以降の16ロングワードが割り当てられている。

trap命令が実行されると68000はスーパーバイザモードに移行し、命令が実行された時点でのpcとsrの値をスーパーバイザスタックに積む。そのうち、該当する例外ベクタの内容を参照し、指定されたアドレスから例外処理を実行する。DOSコールの呼び出しに利用されている未実装命令の実行による例外とは異なり、trap命令による例外処理開始時にスタックに積まれるpcは命令が置かれた直後のアドレスを指しており、小細工をしなく

とも例外処理の最後でrteを実行すればtrap命令のすぐうしろからプログラムの実行を再開できる。

感覚としては“スーパーバイザモードへの移行を伴うサブルーチンコール命令”といったようなもので、その性質上、システムコールを呼び出すのによく用いられている(そのようにシステムが設計される)。本文でも触れたようにX68000ではIOCSの呼び出しにtrap #15を使っている。

このほかX68000+Human68kではtrap #8~#14を内部的に使用している。ふつうのプログラムを作るうえでは知らなくてもすむのだが、興味のある人は「プログラマーズマニュアル」の3.2節末にある参考資料を見てみるとよいだろう。

mes: .dc.b '文字列',0
 という具合だ。ソースプログラム中にIOCSコール番号を生のまま埋め込むのがいやであれば、Human68kのDOSコールの場合のように、インクルードファイルをひとつ作成してその中で、

```

__B__KEYINP      equ    $00
                :
__B__PUTC         equ    $20
__B__PRINT        equ    $21
                :

```

のようにずらずらとIOCSコール番号をシンボル定義しておけばよい。幸いなことにXCにはこのインクルードファイルがIOCSCALL.MACの名前であらかじめ用意されている。また、IOCSCALL.MAC内では、

```

IOCS macro callno
        moveq.l #callno,d0
        trap    #15
        endm

```

というマクロが定義されていて、このマクロとシンボルを利用すると上の例は、

```

move.w #A',d1
IOCS __B__PUTC
とか、
lea.l mes,a1
IOCS __B__PRINT

```

のようにすっきり書けるようになる。今後この連載でIOCSコールを利用するときにはIOCSCALL.MACをインクルードし、このスタイルで記述する(編集部注: 本誌6月号の付録ディスクにも収録させて

いただいたので利用してください)。

実際にIOCSコールを使ったプログラムの一例をリスト1に示す。こんな機会でもなければ誌面に載ることもないようなちっぽけなプログラムLEDOFF.F.Xだ。実行するとすべてのLEDキーをOFF状態にする。AUTOEXEC.BATに潜り込ませるか、Human68k Ver.2ならCONFIG.SYSのPROGRAM=〜行に記述するかして起動時に1回走らせるのが正しい使い方だ。起動直後に“お”とか“ぞ”と打ち込んで“コマンドまたはファイル名が違います”攻撃を受けたことがある人ならLEDOFF.F.Xの有用さに気づいてもらえると思う。

見てのとおりプログラムはLEDキーの状態を操作するIOCSコールLEDMODをループの中から発行するだけという単純さだ。LEDMODは2つのパラメータを採り、d1,lでLEDキーの番号(0~6)、d2,bでONにする(1)かOFFにする(0)かを指定する。リスト1ではループ内でd2を0に固定したままd1を順に変化させてすべてのLEDキーをOFFにしていっている。このことから察しがつくと思うが、IOCSコールでは基本的にd0以外のレジスタの値は保存される(例外はある)。d0だけはIOCSコールの終了ステータスないしは適当な戻り値を返すのに使われる。ちなみに、LEDMODはパラメータの値が範囲外でLEDの設定できなかった場合は-1を、うまく設定できたときは0をd0,lに返す。

そしてマウスへ

とんとんとマウスの話に進む。マウス関連のIOCSコールはコール番号70H~7DHにまとめられており、『プログラマーズマニュアル』を見てもらえばわかるように、X-BASICのマウス操作関数と似たような機能を持ったものがずらっと並んでいる。

X-BASICでマウスを扱ったことがあれば、これらを使いこなすのもわけはない。さっさとサンプルにいつてしまっただろう。リスト2のMSTEST.Sは画面にメニューをひとつ表示し(実際には“終了”という文字列を左上隅に書くだけ)、このメニューの上で左ボタンが押されたら、それに応じた処理をする(終了する)プログラムだ。

11~14行はマウスを使うときには枕詞のように現れる初期化・使用準備の決まりきった手順だ。最初のMS__INITによりマウスカーソルの表示はOFFになり、カーソルパターンは標準の矢印型に、カーソル座標は(0,0)に、カーソルの移動範囲は表示画面の大きさと一致するように初期化される。つづくMS__CURONでマウスカーソルを表示状態にし、SKEY__MODでマウスの右ボタンに割り当てられているソフトウェアキーボードとマウスカーソルの表示/非表示切り換え機能を殺して初期化は完了だ。この3つのIOCSコールの組み合わせは、X-BASICの

```

mouse (0)
mouse (1)

```

リスト0 FILS.H

```

1: *      nameck,files,nfiles用オフセット定義
2:
3:      .offset 0
4: *
5: DRIVE: .ds.b 2      *ドライブ名 'A:'
6: PATH:  .ds.b 64+1   *パス名 '¥BIN¥',0
7: NAME:  .ds.b 18+1   *ファイル名 'ATTRIB',0
8: EXT:   .ds.b 1+3+1  *拡張子 '.X',0
9:      .even
10: NAMBUSIZ:
11: *
12:      .offset 0
13: *
14: FORSYS: .ds.b 21     *システムが使用
15: FATR:   .ds.b 1      *ファイル属性
16: FTIME: .ds.w 1      *ファイル最終更新時刻
17: FDATE: .ds.w 1      *ファイル最終更新日
18: FLEN:  .ds.l 1      *ファイル長
19: PACKEDNAME:
20:      .ds.b 18+1+3+1 *ファイル名
21:      .even
22: FILBUSIZ:
23: *
24:      .text

```

リスト1 LEDOFF.S

```

1: *      全てのLEDキーをOFFにする
2: *
3:      .include      iocSCALL.mac
4:      .include      dosCALL.mac
5: *
6: ent:
7:      moveq.l #0,d2      *OFF
8:      moveq.l #7-1,d1    *LEDキー番号
9: loop: IOCS LEDMOD  *設定
10:      dbra d1,loop      *繰り返す
11:
12:      DOS __EXIT  *終了
13:
14:      .end  ent

```

mouse (4)
にほぼ対応している。

16行からメイン処理が始まる。まず、左ボタンが押されるまで待つ (16~18行)。ボタンの状態を得るにはIOCSコールMS_GETDTを利用する。このIOCSコールはX-BASICのmsstat () に相当し、d0.lの上位ワードにマウスカーソルの相対的な移動量を、下位ワードに左右のボタンの状態を返す。相対的なカーソル移動量のほうはあまり利用されることはないはずだからここでは触れない。

ボタンの状態は第0~7ビットに右ボタン、第8~15ビットに左ボタンのON/OFF状態が返り、ボタンが押されているときは8ビットとも1 (FF_H)、押されていない場合は8ビットとも0 (00_H) になる。右ボタンが押されているかどうかチェックしたければ、

```
IOCS  _MS_GETDT
```

```
tst.b  d0
```

```
beq   押されていない
```

```
押されている~
```

のようにtst.b後のZビットで処理を振り分ければよいのは明らかだろう。左ボタンの場合は、

```
IOCS  _MS_GETDT
```

```
tst.w  d0
```

```
bpl   押されていない
```

```
押されている~
```

という手が使える。どうせ8ビットとも同じ値をとるのだから、第15ビットだけを調べればすむわけだ。

MS_GETDTで左ボタンの押し下げが検出されたら、すかさずMS_CURGTでマウスカーソルの画面上での現在位置を得る (21行)。MS_CURGTはX-BASICのmspos () 関数にあたり、d0.lの上位ワードにマウスカーソルのX座標、下位ワードにY座標を返す。得られた座標がメニュー上にあるかどうかを調べているのが25~28行、やっているのは単純な座標の比較だ。

最後に31行以下が忘れてはならない後始末の処理だ。MS_INITでマウスを再初期化して (マウスカーソルを消し)、SKEY_MODでさっき殺したソフトウェアキーボードを使用可能状態に戻している。

マウスについてはだいたいがリスト2の応用で片がつく。あと、ダブルクリックの判定方法ぐらいは知っていたほうがいいかもしれない。そこでリスト3。リスト2の30行以下と差し換えて使う。ダブルクリックの判定といってもやるべきことは泥臭いといっているほど直接的だ。ボタンが押されたことがわかったら、

1) 一定時間以内に離されるかどうか

2) 一定時間以内にまた押されるかどうか

というチェックを続けて行い、両方に通ったらダブルクリックされたと判断する。これには、IOCSコールのMS_OFTM、MS_ONTMを利用する。d1.wで左右のボタンのどちらか (0なら左、-1なら右)、d2.wで待ち時間を指定し (とくに0のときは

無限と見なされる)、指定時間内にボタンが離されたり (MS_OFTM) 押されたり (MS_ONTM) したら、それまでの経過時間をd0.wに返す。ただし、ドラッグされた場合 (ボタンの状態が変化しないうちにマウスカーソルが動いた場合) にはd0.w=0で即戻ってくる。また、待ち時間を越えた場合はFFFF_Hが返る。待ち時間の単位はなにやらい加減らしく (ループ回数で計時しているのかな)、だいたい40が0.1秒前後に相当する。リスト3では待ち時間を0.2秒程度にするために80を指定してある。

リスト2 MSTEST.S

```
1:      .include      iocscall.mac
2:      .include      doscall.mac
3:      .include      const.h
4:      *
5:  ent:
6:      lea.l    mysp(pc),sp
7:
8:      lea.l    menu(pc),a1      *メニューを描く
9:      IOCS     _B_PRINT      *
10:
11:      IOCS     _MS_INIT      *マウス初期化
12:      IOCS     _MS_CURON     *マウスカーソル表示
13:      moveq.l  #0,d1          *ソフトウェアキーボード
14:      IOCS     _SKEY_MOD     * 表示禁止
15:
16:  loop:  IOCS     _MS_GETDT     *ボタンの状態を得る
17:          tst.w  d0            *左ボタンは押されているか?
18:          bpl    loop          * 押されていないかった
19:
20:          *左ボタンが押された
21:      IOCS     _MS_CURGT     *マウスカーソル座標を得る
22:      move.w   d0,d1          *d1.w = Y座標
23:      swap.w   d0             *d0.w = X座標
24:
25:      cmpi.w   #32,d0         *X座標のチェック
26:      bcc      loop           * 範囲外
27:      cmpi.w   #16,d1         *Y座標のチェック
28:      bcc      loop           * 範囲外
29:
30:          *終了メニュー上だった
31:      IOCS     _MS_INIT      *マウス再初期化
32:      moveq.l  #-1,d1         *ソフトウェアキーボード
33:      IOCS     _SKEY_MOD     * 表示許可
34:
35:      DOS      _EXIT          *終了
36:      *
37:      .data
38:      .even
39:      *
40:  menu:  .dc.b   26,'終了',CR,LF,0
41:      *
42:      .stack
43:      .even
44:      *
45:  mystack:
46:      .ds.l    256
47:  mysp:
48:      .end     ent
```

リスト3 MSTEST2.S

```
30:          *終了メニュー上だった
31:      moveq.l  #0,d1          *左ボタン
32:      moveq.l  #80,d2         *待ち時間 (約0.2秒)
33:      IOCS     _MS_OFTM     *離されるまで待つ
34:      tst.w    d0            *0以下なら
35:      ble      loop          * はしく
36:
37:      IOCS     _MS_ONTM     *押されるまで待つ
38:      tst.w    d0            *0以下なら
39:      ble      loop          * はしく
40:
41:          *ダブルクリックされた
42:      IOCS     _MS_INIT      *マウス再初期化
43:      moveq.l  #-1,d1         *ソフトウェアキーボード
44:      IOCS     _SKEY_MOD     * 表示許可
45:
46:      DOS      _EXIT          *終了
47:      *
48:      .data
49:      .even
50:      *
51:  menu:  .dc.b   26,'終了',CR,LF,0
52:      *
53:      .stack
54:      .even
55:      *
56:  mystack:
57:      .ds.l    256
58:  mysp:
59:      .end     ent
```


お絵かきツールへの応用

最後に応用プログラムとして、簡単なお絵かきツール（グラフィックエディタなんて呼べるほどの代物ではない）を作って今月はおしまいにする。当初はマウスボタンが押されたらその位置に点を打つだけのプログラムにしようと思っていたが、これだとあまりに単純すぎて面白みに欠けるので、IOCSコールで実現できる範囲で多少彩りを添えてみた。

- 1) 色の選択は右ボタンを押すことでポップアップするウィンドウで選べるようにする
- 2) 同じウィンドウ上にはペンパターンのメニューも並べ、複数の中からペンのパターンを選べるようにする（パターンは最大16×16ドット）

一見複雑な処理が要求されそうだが、X68000のハードの機能とIOCSのおかげで、どちらも簡単に実現できる。まず、ウィンドウをポップアップする処理だが、256色2画面の画面モードを使用して、1画面をウィンドウ用、残りを描画用と使い分けることで逃げた。ウィンドウはあらかじめ全部描いておき、X-BASICのvpage関数、home関数に相当するIOCSコールVPAGEとHOMEで表示のON/OFF、表示位置の変更を行う。2点目のペンパターンについては、“外字をSYMBOLで表示する”という手を使った。ペンのパターンを外字に登録しておき、PSETで点を打つ代わりにSYMBOLで描くわけだ。どちらもかなり安直だが、彩りとしての役目は果たしてくれる。

グラフィック関係のIOCSについては約束どおり特に解説しないから『プログラマーズマニュアル』を参照してもらいたい。一応リスト4にLINEのサンプルを示しておく。COMMAND.X上からグラフィック画面に直線を描画するプログラムだ。7行のパラメータの個数と、23行のIOCSコール番号を変更すればBOXやFILL、CIRCLEにも対応できるので気が向いたら試してみしてほしい。あまり使い道のないプログラムだが、派手なバッチファイルを作りたいときなんかには利用できるだろう。

なお、コマンドラインで指定された数字（の文字列）を数値に変換するのにリスト5中のサブルーチンatoiを利用しているので、実行ファイル作成時にはこれも忘れずにリンクすること。このatoiは今後も使うことがあるかもしれない（変に凝ってしまったのであまりよいわけではないが）。また、LINE.Xはグラフィック画面の初期化を行わないので、使用時にはSCREENコマンドであらかじめグラフィック画面を使用可能に設定しておく必要がある。

atoiについて1点だけ補足しておく。5～8行ではCフラグを反転（0 ↔ 1）するマクロCCFを定義している。その実体は、

```
eori.w #1, ccr
```

というオペランドにccrが登場するという見慣れない命令だ。この命令は任意のフラグを反転するのに

使う。排他的論理和の意味と、ccrの構造を思い出してもらいたい。同様の命令としては、

```
andi.w #n, ccr
```

```
ori.w #n, ccr
```

があり、それぞれ、ccrレジスタ中の任意のフラグをリセットしたりセットしたりするのに用いられる。

STAMP.Sの解説

では、手抜きいっぱいのお絵かきプログラム、リスト6のSTAMP.Sを見てもらおう。比較的読みやすく書けたと思うので、これまでの話のまとめのつもりで読んでみてもらいたい。各ルーチンごとにポイントとなる部分を拾って軽く解説しておく。

●エントリ～終了（62行～）

Interruptスイッチなどによってプログラムの実行が中断された場合に後始末をせずに親プロセスに帰るのがいやだったので、67～72行で前回のASX.Xとまったく同じ手順で中断時の戻りアドレスを77行のラベルbreakの位置に設定している。

break以降では諸々の後始末をするサブルーチン呼び出ししてから、キーバッファをクリアし、exitで実行終了する。マウスしか使わないプログラムでキーバッファを気にしているのが変に見えるかもしれないが、“マウスしか使わないからこそ”この処理が必要なのだ。これを怠ると、プログラム走行中に誤って押されたキーがプログラム終了後にまとめて吐き出されることになる。

●初期化ルーチン（275行～）

278～290行でDOSコールconctrlによって画面モードを横512×縦512ドット、256色モードに切り換えたうえで、邪魔なファンクションキー行とカーソルを消している。画面モードとファンクションキー行についてはあとで元に戻せるように（374行以下の後始末ルーチン参照）現在の状況をワークエリアにしまっておく。それが作法というものだ。あと、このサブルーチンでは頭でlinkし、リターンする直前でunlkすることによってDOSコール呼び出し時のスタック補正を省略するという姑息なテクニックが使われている。あまり褒められたことではないが、一度やって見せたかった。

293、294行は下位のサブルーチン呼び出して、ペンパターンとして利用する外字の定義を行っている。ここでも、あとで元に戻せるように現在の外字の定義を取得・待避しておくのを忘れない。定義する外字のフォントパターンは436行以下に用意しており、16ワードが1文字分のデータにあたる。

頭に縦横のドット数がつけてあるのはほかとの兼ね合いで、実際には使っていない。フォントパターンは438～453行の最初の1個だけは見やすく2進数で表記してみた（2個目以降はスペースの都合で詰めて16進数で表記してある）。これを見ればフォントパターンの形式・作り方は一目瞭然だろう。

●メニューウィンドウの初期化（309行～）

前述のとおり、メニューはあらかじめ全部描いておく。描画に必要なデータはデータセクションに用意しておき、これを次々にIOCSコールに渡している。

●メイン処理 (88行～)

多少冗長な作りになっているが、マウスのボタンの状態をチェックし、ボタンが押されていたならその位置に応じてそれなりの処理を行うというパターンの組み合わせであり、リスト2と基本的には大差ない。左ボタンが押された場合は、まずメニューウィンドウ上かどうかを調べ、ウィンドウ外（もしくはウィンドウが非表示状態）であれば197行に飛んでSYMBOLで現在設定されているペンパターン（に対応する外字1文字）を描く。ウィンドウ上だった場合は、マウスカーソル座標から、

- 1) ペン選択メニュー上
- 2) 色選択メニュー上
- 3) 終了メニュー上
- 4) いずれでもないウィンドウの外枠

を識別し、対応する処理を行う。1), 2) の場合はさらにメニュー上のどの部分かの判定が加わることになる。また、ウィンドウの外枠で左ボタンが押された場合はウィンドウをドラッグするようにしてみた。本来ならマウスの動きに連動してリアルタイムでウィンドウの位置を変更することもできたのだが、もっと単純に、ボタンが離された位置へいきなりウィンドウを移動するようになっている。ここは読者に手を入れてもらいたい部分のひとつだ。

左ボタンの処理に比べれば、207行以下の右ボタンによるメニューウィンドウのON/OFF切り換え処理はシンプルだ。現在メニューが表示中かどうかを覚えておくワークmenuflagを調べて (209行)、もしメニューがすでに表示中であれば212行以下でVPAGEによりメニューが描かれているページを

非表示にする。メニューが表示されていなければ221行以下で現在のマウスカーソルの位置にメニューを表示する。

なお、222行でmenuflagをセットするのに使っているst.bは、任意の1バイトをFF_Hにする命令だ（オペランドサイズはバイト固定）。正確にはstの一般形はsXX（sはSetの略）であり、XXの部分には条件分岐命令同様の条件が入る。sXXは命令実行の時点でこの条件が成り立っていればオペランドをFF_Hにし、条件が成り立っていなければ00_Hにする命令で、stはこの条件が“t (always True: 常に真)”になった形だ。条件が常に成り立つわけだから、オペランドを00_Hにすることはありえない。逆にsfという命令は条件が“f (always False: 常に偽)”であり、任意の1バイトを00_Hにするのに使える。趣味の問題だが、人によってはclr.bの代わりに使うこともある。

*

『プログラマーズマニュアル』をパラパラと眺めると、それ単体でプログラムとして成り立つようなIOCSコールがいくつか見つかると思う。例を挙げるなら、コール番号7F_HのONTIME（本体を立ち上げてからの時間を100分の1秒単位で返す）とか8E_HのBOOTINF（前面の電源スイッチにより起動されたのか、タイマにより起動されたのか、また、どのデバイスから起動されたのかといったブート情報を返す）なんかは、IOCSコールからの戻り値を表示するだけでもそれなりに役にたつ（ことがあるかもしれない）プログラムになる。この類のプログラムはあって困るものでもなし、暇を見つけて作っておくとよいだろう。

来月は、グラフィックをもう少し本格的に取り上げる予定でいる。

リスト4 LINE.S

```

1:      .include      doscall.mac
2:      .include      iocscall.mac
3:      .include      const.h
4:      *
5:      .xref      atoi      *外部参照
6:      *
7:      PARCNT      equ      6      *IOCSに渡すパラメータの個数
8:      *
9:      .text
10:     .even
11:     *
12:     ent:
13:     *
14:     lea.l      mysp(pc),sp      *spを初期化する
15:     *
16:     bsr      getpar      *パラメータを取得する
17:     *
18:     moveq.l      #-1,d1      *グラフィック画面は
19:     IOCS      _APAGE      * 初期化されているか?
20:     tst.b      d0
21:     bmi      error      *未初期化ならエラー終了
22:     *
23:     lea.l      giocspar(pc),a1      *直線描画
24:     IOCS      _LINE      *
25:     tst.b      d0      *エラー?
26:     bmi      usage      *パラメータの値が変
27:     DOS      _EXIT      *正常終了
28:     *
29:     *
30:     *      PARCNT個の数値をバッファにセットする
31:     *
32:     getpar:
33:     tst.b      (a2)+      *空文字列なら
34:     beq      usage      * 使用法を表示して終了
35:     *
36:     lea.l      giocspar(pc),a1      *a1=パラメータ格納バッファ
37:     moveq.l      #PARCNT-1,d1      *d1=ループカウンタ
38:     getpr0:      move.l      a2,-(sp)      *文字列→数値変換
39:     bsr      atoi      *
40:     movea.l      (sp)+,a2      *a2=続く文字列
41:     bmi      usage      *うまく変換できなかった
42:     move.w      d0,(a1)+      *パラメータを格納

```

```

43:     dbra      d1,getpr0      *PARCNT回繰り返す
44:     *
45:     rts
46:     *
47:     *
48:     *      使用法の表示&エラー終了
49:     *
50:     usage:
51:     move.w      #STDERR,-(sp)      *標準エラー出力へ
52:     pea.l      usgmes(pc)      * メッセージを
53:     DOS      _FPUTS      * 出力する
54:     addq.w      #6,sp
55:     *
56:     error:      move.w      #1,-(sp)      *終了コード1を持って
57:     DOS      _EXIT2      * エラー終了
58:     *
59:     *
60:     *      データ&ワーク
61:     *
62:     .data
63:     .even
64:     *
65:     usgmes:      .dc.b      '機 能:グラフィック画面に直線を描きます'
66:     CR,LF
67:     .dc.b      '使用法:LINE X0 Y0 X1 Y1'
68:     .dc.b      ' パレットコード ラインスタイル'
69:     CR,LF,0
70:     *
71:     .bss
72:     .even
73:     *
74:     giocspar:
75:     .ds.w      PARCNT      *パラメータバッファ
76:     *
77:     .stack
78:     .even
79:     *
80:     mystack:
81:     .ds.l      256      *スタック領域
82:     mysp:
83:     .end      ent

```


リスト5 ATOI.S

```

1: .include const.h
2: *
3: .xdef atoi
4: *
5: CCF macro dreg #C000001,ccr *Cビットを反転するマクロ
6: eor.w #X000001,ccr
7: *
8: endm
9: *
10: TOUPPER macro dreg *英小文字→大文字変換マクロ
11: local skip
12: cmpi.b #'a',dreg
13: bcs skip
14: cmpi.b #'z'+1,dreg
15: bcc skip
16: subi.b #'a'-'A',dreg
17: skip:
18: endm
19: *
20: .text
21: .even
22: *
23: *atoi(str)
24: *機能: 数値を表す文字列を16ビット符号付整数に変換する
25: *戻り値: d0.w = 変換された値
26: * (sp).l = 続く文字列へのポインタ
27: * N = 1文字も変換できなかった場合に1
28: *メモ: 文字列先頭に余分な空白を置くことを許す
29: * 先頭に'+','-'の符号をつけてもよい
30: * ex) 123, +123, -123
31: * '$','x','X'をつけるなど16進数とみなす
32: * ex) $12AB, -$12AB, x12AB, X12AB
33: *
34: str = 8
35: *
36: atoi:
37: link a6,#0 *スタックフレーム生成
38: movem.l d1-d3/a0,-(sp) *レジスタ待避
39: movea.l str(a6),a0 *a0=文字列へのポインタ
40: bra atoi1
41: *
42: atoi0: addq.w #1,a0 *文字列先頭の空白を
43: atoi1: cmpi.b #SPACE,(a0) *飛ばす
44: beq atoi0
45: cmpi.b #TAB,(a0)
46: beq atoi0
47: *
48: moveq.l #1,d2 *d2=符号(+)
49: cmpi.b #'+',(a0) * '+'が指定されたか?
50: beq atoi2
51: cmpi.b #'-',(a0) * '-'が指定されたか?
52: bne atoi3
53: moveq.l #-1,d2 *d2=符号(-)
54: atoi2: addq.w #1,a0 *符号の分ポインタを進める
55: *
56: atoi3: moveq.l #0,d0 *結果を返すd0をクリア
57: moveq.l #0,d1 *作業用のd1をクリア
58: moveq.l #-1,d3 *仮にエラーフラグを立てる
59: *
60: cmpi.b #'$',(a0) *16進の指定かどうか調べず
61: beq htoi
62: cmpi.b #'X',(a0)
63: beq htoi
64: cmpi.b #'x',(a0)
65: beq htoi

```

```

66:
67: bra atoi5 *10進文字列のとき
68: *
69: atoi4: addq.w #1,a0
70: atoi5: move.b (a0),d1 *1文字取り出す
71: bsr isdigit *数字か?
72: bcs atoiq *そうでなければ終了
73: mulu.w #10,d0 *10進1桁分左にシフト
74: swap.w d0 *上位ワードが
75: tst.w d0 *0でなければ
76: bne atoi6 *オーバーフローした
77: swap.w d0
78: add.w d1,d0 *下位に1桁追加
79: moveq.l #0,d3
80: bra atoi4 *エラーフラグをクリア
81: *
82: *16進文字列のとき
83: htoi: addq.w #1,a0
84: move.b (a0),d1 *1文字取り出す
85: bsr isxdigit *16進数値か?
86: bcs atoiq *そうでなければ終了
87: asl.l #4,d0 *16進1桁分左にシフト
88: swap.w d0 *上位ワードが
89: tst.w d0 *0でなければ
90: bne atoi6 *オーバーフローした
91: swap.w d0
92: add.w d1,d0 *下位に1桁追加
93: moveq.l #0,d3
94: bra htoi *エラーフラグをクリア
95: *
96: atoi6: moveq.l #-1,d3 *オーバーフローが発生
97: atoiq: muls.w d2,d0 *符号を掛けて最終結果を得る
98: move.l a0,str(a6) *続く文字列へのポインタを返す
99: tst.w d3 *エラーフラグをccrに反映する
100: movem.l (sp)+,d1-d3/a0 *レジスタ復帰
101: unlk a6 *スタックフレーム解放
102: rts
103: *
104: *
105: * 10進文字→数値変換 (d1.b)
106: * (エラー時はC=1)
107: isdigit:
108: subi.b #'0',d1
109: bcs isdgtq
110: cmpi.b #9+1,d1
111: CCF
112: isdgtq: rts
113: *
114: *
115: * 16進文字→数値変換 (d1.b)
116: * (エラー時はC=1)
117: isxdigit:
118: TOUPPER d1
119: subi.b #'0',d1
120: bcs isxdgq
121: cmpi.b #9+1,d1
122: CCF
123: bcc isxdgq
124: subq.b #'A'-'0'-10,d1
125: bcs isxdgq
126: cmpi.b #15+1,d1
127: CCF
128: isxdgq: rts
129: *
130: .end

```

リスト6 STAMP.S

```

1: .include doscall.mac
2: .include iocscall.mac
3: *
4: CFKEYMOD equ 14 *CONCTRLモード番号
5: CSCREEM equ 16
6: CCURON equ 17
7: CCUROFF equ 18
8: *
9: HIDEFKEY equ 3 *ファンクションキー行非表示
10: DOS_GM3 equ 4 *画面モード512x512,256色
11: *
12: DISABLESKEY equ 0 *ソフトウェアキーボード禁止
13: ENABLESKEY equ -1 *ソフトウェアキーボード許可
14: *
15: WINH equ 272 *メニューウィンドウ幅
16: WINV equ 104 *メニューウィンドウ高さ
17: *
18: USERPAGE equ 1 *描画を行う画面
19: BIT_USERPAGE equ #0010
20: MENUPAGE equ 0 *メニューを表示する画面
21: BIT_MENUPAGE equ #0001
22: *
23: * *メニュー表示
24: SHOWMENU equ BIT_USERPAGE|BIT_MENUPAGE
25: * *メニュー非表示
26: HIDEMENU equ BIT_USERPAGE
27: *
28: GAIJITOP equ $eb9f *全角外字の先頭文字コード
29: FONTI6 equ $0008 *8x16,16x16
30: *
31: PATMAX equ 8 *ペンパターンの最大数
32: *
33: *
34: *グラフィック関係IOCSデータ受け渡し領域の構造
35: *
36: .offset 0
37: *
38: X0: .ds.w 1 *POINT *FILL *BOX
39: Y0: .ds.w 1
40: RETCOL:

```

```

41: X1: .ds.w 1
42: POINTBUFSIZ:
43: Y1: .ds.w 1
44: COL: .ds.w 1
45: FILLBUFSIZ:
46: LS: .ds.w 1
47: BOXBUFSIZ:
48: *
49: *
50: * フォント読み込み領域の構造
51: *
52: .offset 0
53: *
54: XLEN: .ds.w 1
55: YLEN: .ds.w 1
56: FPAT: .ds.w 16 *16x16
57: FNTBUFSIZ:
58: *
59: .text
60: .even
61: *
62: ent:
63: lea.l mysp(pc),sp *spを初期化する
64: *
65: bsr init *画面などの初期化
66: *
67: pea.l break(pc) *中断時の戻りアドレスを設定
68: move.w #_CTRLVC,-(sp)
69: DOS _INTVCS
70: move.w #_ERRJVC,(sp)
71: DOS _INTVCS
72: addq.l #6,sp
73: *
74: bsr setupmenu *メニューウィンドウの初期化
75: bsr main *メイン処理
76: *
77: break: bsr windup *後始末
78: *
79: move.w #-1,-(sp) *キーバッファクリア
80: DOS _KFLUSH

```

```

81:      addq.l  #2,sp      *
82:
83:      DOS      _EXIT      *終了
84:
85:  *
86:  *      メイン処理
87:  *
88:  main:
89:      IOCS      _MS_GETDT      *ボタンの状態を取得
90:      tst.b     d0      *右ボタンが押されている?
91:      bne      rdown
92:      tst.w     d0      *左ボタンが押されている?
93:      bpl      main
94:  *
95:  ldown:      *左ボタンが押された
96:      IOCS      _MS_CURGT      *マウスカーソル位置を取得
97:      move.w    d0,d1      *d1.w = y
98:      clr.w     d0
99:      swap.w    d0      *d0.w = x
100:     tst.b     menuflag      *ウィンドウは表示中か?
101:     beq       pset
102:
103:     move.w     winx(pc),d2      *d2.w = ウィンドウ表示位置X
104:     move.w     winy(pc),d3      *d3.w = ウィンドウ表示位置Y
105:
106:     cmp.w     d2,d0      *ウィンドウ上かどうかチェック
107:     bcs      pset
108:     cmp.w     d3,d1
109:     bcs      pset
110:     addi.w    #WINH,d2
111:     addi.w    #WINV,d3
112:     cmp.w     d2,d0
113:     bcc      pset
114:     cmp.w     d3,d1
115:     bcc      pset
116:
117:     *ウィンドウ内でクリックされた
118:     sub.w     winx(pc),d0      *d0.w = ローカル座標
119:     sub.w     winy(pc),d1      *d1.w = ローカル座標
120:     move.w    d0,pntbuf+X0      *x,yそれぞれを待避しておく
121:     move.w    d1,pntbuf+Y0
122:
123:     subq.w    #8,d0      *ウィンドウの左余白
124:     bcs      drag
125:     subq.w    #8,d1      *ウィンドウの上余白
126:     bcs      drag
127:     cmp.w     #256,d0      *ウィンドウの右余白
128:     bcc      drag
129:     cmp.w     #16,d1
130:     bcc      ldown1
131:
132:     *上陸のメニュー内
133:     cmp.w     #224,d0
134:     bcs      ldown0
135:
136:     *終了ボックス内
137:     rts      *メインループを抜ける
138:
139:     ldown0:  sub.l.w    #32,d0      *ヘンメニューより左
140:     bcs      drag
141:     divu.w    #21,d0
142:     swap.w    d0
143:     cmpi.w    #16,d0      *ヘンハタンの隙間の余白
144:     bcc      drag
145:     swap.w    d0
146:  *
147:  *      selpen:
148:     *ヘンメニュー内
149:     move.w    d0,d2      *d2.w = ヘン番号
150:     addi.w    #GAIJITOP,d0      *新パターンを設定
151:     move.w    d0,curpat
152:
153:     moveq.l   #MENUPAGE,d1      *メニュー用ページに
154:     IOCS      _APAGE      *切り換える
155:
156:     lea.l     curpen(pc),a1      *以前の枠を消す
157:     move.w    #255,COL(a1)      *
158:     IOCS      _BOX
159:
160:     mulu.w    #24,d2      *新たに枠を描く
161:     addi.w    #38,d2
162:     move.w    d2,X0(a1)
163:     addi.w    #19,d2
164:     move.w    d2,X1(a1)
165:     move.w    #1,COL(a1)
166:     IOCS      _BOX
167:
168:     bsr      lwait      *左ボタンが離されるのを待つ
169:
170:     bra      ldown2      *描画用ページに戻す
171:
172:     ldown1:  sub.l.w    #24,d1      *ヘンメニューと
173:     bcs      drag      *色メニューの隙間の余白
174:     cmpi.w    #64,d1
175:     bcc      drag      *ウィンドウの下余白
176:
177:     *色メニュー内
178:     moveq.l   #MENUPAGE,d1      *メニュー用ページに
179:     IOCS      _APAGE      *切り換える
180:
181:     lea.l     pntbuf(pc),a1      *マウスカーソル位置から
182:     IOCS      _POINT      *色を拾う
183:     move.w    RETCOL(a1),d0
184:     move.w    d0,curcol      *カレントカラーにセット
185:
186:     lea.l     coldat(pc),a1      *カレントカラーで
187:     move.w    d0,COL(a1)      *メニュー左上の枠を
188:     IOCS      _FILL      *塗り潰す
189:
190:     ldown2:  moveq.l   #USERPAGE,d1      *描画用ページに戻す
191:     IOCS      _APAGE
192:
193:     bra      main      *メインループへ
194:
195:  drag:      *ウィンドウの外枠でクリックされた
196:     bsr      lwait      *左ボタンが離されるのを待つ
197:     bra      menuon      *ウィンドウを描き直す

```

```

196:  *
197:  pset:      *ウィンドウ外でクリックされた
198:     lea.l     setdat(pc),a1      *マウスカーソルの位置に
199:     subq.w    #8,d0      *パターンを描く
200:     move.w    d0,X0(a1)
201:     subq.w    #8,d1
202:     move.w    d1,Y0(a1)
203:     IOCS      _SYMBOL
204:
205:     bra      main      *メインループへ
206:
207:  rdown:      *右ボタンが押された
208:     move.l     #(WINH/2)<<16,ofst      *表示位置オフセット
209:     tst.b     menuflag      *メニューは表示中か?
210:     beq       menuon
211:
212:     *メニューウィンドウを消す
213:     clr.b     menuflag      *フラグをリセット
214:     moveq.l   #HIDEMENU,d1      *メニューページ非表示
215:     IOCS      _VPAGE
216:
217:     bsr      rwait      *右ボタンが離されるのを待つ
218:
219:     bra      main      *メインループへ
220:
221:  *
222:  menuon:      *メニューウィンドウを出す
223:     st.b     menuflag      *フラグを立てる
224:     IOCS      _MS_CURGT      *マウスカーソル位置を取得
225:     move.w    d0,d1      *d1.w = y
226:     swap.w    d0      *d0.w = x
227:
228:     sub.w     ofst+X0(pc),d0      *ウィンドウが
229:     bcc      mon0      *画面からはみ出さないよう
230:     clr.w     d0      *調整する
231:     sub.w     ofst+Y0(pc),d1
232:     bcc      mon1
233:     clr.w     d1
234:     cmp.w     #512-WINH,d0
235:     bcs      mon2
236:     move.w    #512-WINH,d0
237:     cmp.w     #512-WINV,d1
238:     bcs      mon3
239:     move.w    #512-WINV,d1
240:
241:  mon3:      move.w    d0,winx      *表示位置を格納
242:     move.w    d1,winy
243:
244:     moveq.l   #0,d2      *ウィンドウを目的の位置へ
245:     sub.w     d0,d2      *移動する
246:     andi.w    #511,d2
247:     moveq.l   #0,d3
248:     sub.w     d1,d3
249:     andi.w    #511,d3
250:     moveq.l   #BIT_MENUPAGE,d1
251:     IOCS      _HOME
252:
253:     moveq.l   #SHOWMENU,d1      *メニューページ表示
254:     IOCS      _VPAGE
255:
256:     bsr      rwait      *右ボタンが離されるのを待つ
257:
258:     bra      main      *メインループへ
259:
260:  *
261:  rwait:      *右ボタンが離されるのを待つ
262:     IOCS      _MS_GETDT
263:     tst.b     d0
264:     bne      rwait
265:     rts
266:
267:  *
268:  lwait:      *左ボタンが離されるのを待つ
269:     IOCS      _MS_GETDT
270:     tst.w     d0
271:     bmi      lwait
272:     rts
273:
274:  *
275:  init:      *初期化
276:     link      a6,#0
277:
278:     *画面
279:     move.w    #DOS_GM3,-(sp)      *画面を512x512,256色に
280:     move.w    #CSCREEN,-(sp)      *初期化
281:     DOS      _CONCTRL
282:     move.w    d0,scrnsav      *現在の画面モードを待避
283:
284:     move.w    #HIDEKEY,-(sp)      *ファンクションキー行を
285:     move.w    #CFKEYMOD,-(sp)      *非表示に設定
286:     DOS      _CONCTRL
287:     move.w    d0,fkeymsav      *現在のファンクションキー行
288:     *モードを待避
289:
290:     move.w    #CCUROFF,-(sp)      *カーソル非表示モード
291:     DOS      _CONCTRL
292:
293:     *外字
294:     bsr      savfont      *待避
295:     bsr      deffont      *定義
296:
297:     *マウス
298:     IOCS      _MS_INIT      *マウス初期化
299:     IOCS      _MS_CURON      *マウスカーソル表示
300:     moveq.l   #DISABLESKEY,d1      *ソフトウェアキーボード
301:     IOCS      _SKEY_MOD      *表示禁止
302:
303:     unlk      a6
304:     rts
305:
306:  *
307:  *      メニューの初期化
308:  *      (あらかじめ全部描いておく)
309:
310:  setupmenu:
311:     moveq.l   #MENUPAGE,d1      *メニュー用ページに

```



```

311:      IOCS      _APAGE      * 切り換える
312:
313:      moveq.l   #HIDEMENU,d1  *メニュー用ページ非表示
314:      IOCS      _VPAGE      *
315:
316:      lea.l     fildat(pc),a1  *ウィンドウ枠を塗り潰す
317:      IOCS      _FILL      *
318:
319:      lea.l     boxes(pc),a1   *BOXを必要なだけ描く
320:      boxlp:    tst.w        (a1)
321:               bmi         boxed
322:               IOCS      _BOX
323:               lea.l     BOXBUFSIZ(a1),a1
324:               bra      boxlp
325:      boxed:
326:               lea.l     mendat(pc),a1
327:               IOCS      _SYMBOL
328:               bsr      makecoltbl
329:               bsr      makecoltbl
330:
331:      moveq.l   #USERPAGE,d1  *描画用ページに切り換える
332:      IOCS      _APAGE      *
333:
334:      clr.b     menuflag      *フラグをリセット
335:
336:      rts
337:
338:      *
339:      *      256色の色テーブルを描く
340:      *
341:      makecoltbl:
342:      link      a6,#-FILLBUFSIZ
343:
344:      lea.l     -FILLBUFSIZ(a6),a1  *a1=FILL用パラメータ領域
345:
346:      moveq.l   #0,d1           *d1=色
347:
348:      move.w     #32,Y0(a1)      *(8,32)-(8+7,32+7)から
349:      move.w     #32+7,Y1(a1)    *
350:
351:      moveq.l   #8-1,d6         *縦に8個
352:      clp0:     move.w     #8,X0(a1)
353:               move.w     #8+7,X1(a1)
354:
355:      moveq.l   #32-1,d7         *横に32個
356:      clp1:     move.w     d1,COL(a1)
357:               IOCS      _FILL
358:               addq.w     #8,X0(a1)
359:               addq.w     #8,X1(a1)
360:               addq.w     #1,d1
361:               dbra      d7,clp1
362:               *横1列分繰り返す
363:
364:               addq.w     #8,Y0(a1)
365:               addq.w     #8,Y1(a1)
366:               dbra      d6,clp0
367:               *下に8ドット移動
368:               *繰り返す
369:               unlk      a6
370:               rts
371:
372:      *      後始末
373:      *
374:      windup:
375:      link      a6,#0
376:
377:      move.w     scrnmsav,-(sp)   *画面モードを戻す
378:      move.w     #CSCREEN,-(sp)
379:      DOS      _CONCTRL
380:
381:      move.w     fkeymsav,-(sp)  *ファンクションキー一行の
382:      move.w     #CFKEYMOD,-(sp) *モードを戻す
383:      DOS      _CONCTRL
384:
385:      move.w     #CCURON,-(sp)   *カーソル表示モード
386:      DOS      _CONCTRL
387:
388:      bsr      rstfont
389:
390:      IOCS      _MS_INIT      *マウス初期化
391:      moveq.l   #ENABLESKEY,d1  *ソフトウェアキーボード
392:      IOCS      _SKEY_MOD      *表示許可
393:
394:      unlk      a6
395:      rts
396:
397:      *
398:      *      外字の先頭8文字のフォントパターンを待避する
399:      *
400:      savfont:
401:      lea.l     fontbuf(pc),a1
402:      move.l     #FONT16<<16|GAIJITOP,d1
403:      moveq.l   #PATMAX-1,d2
404:      savlp:    IOCS      _FNTGET
405:               addq.w     #1,d1
406:               lea.l     FNTBUFSIZ(a1),a1
407:               dbra      d2,savlp
408:               rts
409:
410:      *
411:      *      外字の先頭8文字にフォントパターンを設定する
412:      *
413:      deffont:
414:      lea.l     fontdat+FPAT(pc),a1
415:      defnt0:   move.l     #FONT16<<16|GAIJITOP,d1
416:               moveq.l   #PATMAX-1,d2
417:               IOCS      _DEFCHR
418:               addq.w     #1,d1
419:               lea.l     FNTBUFSIZ(a1),a1
420:               dbra      d2,deflp
421:               rts
422:
423:      *
424:      *      savfontで待避したフォントパターンを復帰する

```

```

425:      *
426:      rstfont:
427:      lea.l     fontbuf+FPAT(pc),a1
428:      bra      defnt0
429:
430:      *
431:      *      データ&ワーク
432:      *
433:      .data
434:      .even
435:      *
436:      fontdat:
437:      .dc.w     16,16
438:      .dc.w     $0000000000000000
439:      .dc.w     $0000000000000000
440:      .dc.w     $0000000000000000
441:      .dc.w     $0011111101111111
442:      .dc.w     $0010000001010001
443:      .dc.w     $0001000001010010
444:      .dc.w     $0001000001011000
445:      .dc.w     $0000100000101000
446:      .dc.w     $0000100000100000
447:      .dc.w     $0000010000010000
448:      .dc.w     $0000010000010000
449:      .dc.w     $0000101000001000
450:      .dc.w     $0001101000000100
451:      .dc.w     $0010010100000100
452:      .dc.w     $0100010100000010
453:      .dc.w     $0111111011111110
454:
455:      .dc.w     16,16
456:      .dc.w     $0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
457:      .dc.w     $0080,$0000,$0000,$0000,$0000,$0000,$0000,$0000
458:
459:      .dc.w     16,16
460:      .dc.w     $0000,$0000,$0000,$0000,$0000,$0000,$0000,$0180
461:      .dc.w     $0180,$0000,$0000,$0000,$0000,$0000,$0000,$0000
462:
463:      .dc.w     16,16
464:      .dc.w     $0000,$0000,$0000,$0000,$0000,$0000,$0000,$01c0
465:      .dc.w     $01c0,$01c0,$0000,$0000,$0000,$0000,$0000,$0000
466:
467:      .dc.w     16,16
468:      .dc.w     $0000,$0000,$0000,$0000,$0000,$0410,$0220,$0140
469:      .dc.w     $0080,$0140,$0220,$0410,$0000,$0000,$0000,$0000
470:
471:      .dc.w     16,16
472:      .dc.w     $0000,$0000,$0000,$0000,$0000,$00c0,$0600,$0300
473:      .dc.w     $0180,$00c0,$0060,$0030,$0000,$0000,$0000,$0000
474:
475:      .dc.w     16,16
476:      .dc.w     $0000,$03e0,$07f0,$0ff8,$1ffc,$3ffe,$7fff,$7fff
477:      .dc.w     $7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff
478:
479:      .dc.w     16,16
480:      .dc.w     $0000,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff
481:      .dc.w     $7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff,$7fff
482:
483:      fildat:   .dc.w     0,0
484:      .dc.w     WINH-1,WINV-1
485:      .dc.w     255
486:
487:      boxes:
488:      box1:    .dc.w     2,2,WINH-2-1,WINV-2-1,1,$ffff
489:      box2:    .dc.w     6,6,33,25,1,$ffff
490:      curpen:
491:      box3:    .dc.w     38,6,57,25,1,$ffff
492:      box4:    .dc.w     230,6,265,25,1,$ffff
493:      box5:    .dc.w     6,30,265,97,1,$ffff
494:
495:
496:      mendat:   .dc.w     40,8
497:      .dc.l     patstr
498:      .dc.b     1,1
499:      .dc.w     1
500:      .dc.b     1,0
501:
502:      coldat:   .dc.w     8,8,31,23,255
503:
504:      setdat:   .dc.w     0,0
505:      .dc.l     curpat
506:      .dc.b     1,1
507:      curcol:   .dc.w     255
508:      .dc.b     1,0
509:
510:      curpat:   .dc.b     $eb,$9f,0
511:
512:      patstr:   .dc.b     $eb,$9f,$20,$eb,$a0,$20
513:      .dc.b     $eb,$a1,$20,$eb,$a2,$20
514:      .dc.b     $eb,$a3,$20,$eb,$a4,$20
515:      .dc.b     $eb,$a5,$20,$eb,$a6,$20
516:      .dc.b     '終了',0
517:      *
518:      .bss
519:      .even
520:      *
521:      fontbuf:   .ds.b     FNTBUFSIZ*8
522:      ofst:
523:      pntbuf:   .ds.b     POINTBUFSIZ
524:      winx:     .ds.w     1
525:      winy:     .ds.w     1
526:      scrnmsav: .ds.w     1
527:      fkeymsav: .ds.w     1
528:
529:      menuflag: .ds.b     1
530:      *
531:      .stack
532:      .even
533:      *
534:      mystack:   .ds.l     1024
535:
536:      msp:
537:      .end      ent

```

PASCALのデータ型を見る

Fujii Yoshimi/Fujiki Takeshi

藤井義巳/藤木健士

PASCALはさまざまなデータを多彩な方法で取り扱うことができます。それは整数や実数などの数値、文字列といったものから集合やポインタにまで及びます。それではPASCAL言語におけるデータの扱い方をまとめて見てみましょう。

連載も3回目になりますが、読者の皆さんはもうPASCALのプログラムをいくつか書いてみられたことと思います。Cを知っている人なら「なあんだ、簡単じゃねーか」と思われたでしょう。Cを知らない人でもそれほど難しくありません。ただ、PASCALは型の厳しい言語なので、型についてよく知っておかないとしょっちゅうコンパイラから文句をいわれます。たとえばCなら整数変数に実数値を代入しても、勝手に変換してくれていたのに、PASCALではエラーになるといった具合です。そこで今回はその型について、少し詳しく説明することになります。

データ型

WirthはPASCALをプログラミングの教育に使用したいと考えました。彼は著書『アルゴリズム+データ構造=プログラム』で、プログラムを作る際のデータ構造の大切さを教えています。その彼が設計したPASCALが豊富なデータ型を備えていたのは当然のことで、さまざまなデータ構造を直接に記述することができます。PASCALのデータ型はおおまかに単純データ型と構造データ型、およびポインタ型に分類することができます。単純データ型はさらに、整数型、実数型、列挙型、論理型、文字型、部分範囲型に分かれます(図1)。また、実数型以外の単純データ型は順序型とも呼ばれて、共通の特徴を持っています。構造データ型には配列型、ファイル型、集合型、レコード型があります。PASCALの構造データ型はPackedという形容詞をつけると、多少の速度を犠牲にしても主記憶を食わないように、詰め込んだデータ型になります。このあたりは処理系によって対応がまちまちですが、後述の文字列型に関しては必ずPackedと書かなければならないことになっています。

PASCALがデータ構造の表現力に優れているのは、レコード型とポインタ型のおかげです。リスト構造、ツリー構造、キューなどの基本的なデータ構造をレコード型とポインタなしで表現することを想像してみてください。なにを隠そうFORTRANの世界では、21世紀を迎えようとする今日になっても、「データ構造はすべて配列で作る」なんて野蛮なことが行われているのです。信じられませんか。

前置きはこれくらいにして、それぞれの型について説明していきましょう。

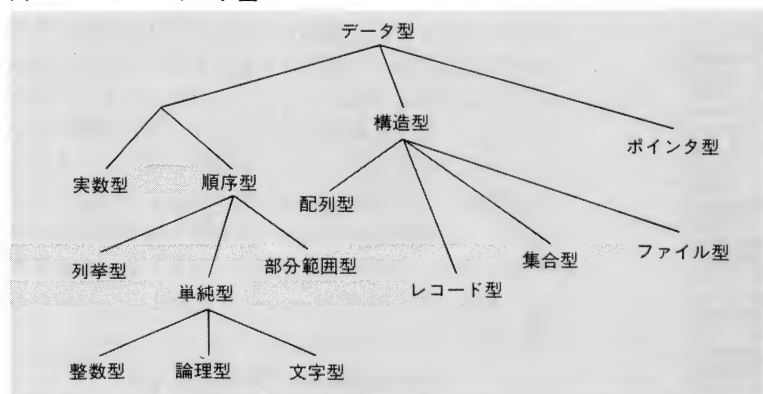
整数型と実数型

整数型と実数型はどの言語でも馴染みの型ですね。PASCALの整数型の最大値は定義済み定数Max Intで知ることができます。例によって8086系CPUの処理系ではMax Intは32767であることが多く、PurePASCALは $2^{31}-1$ (計算して!)です。実数についてはなにもいうことはありません。式の中で実数と整数は混在して使うことが許されています。というより、整数は実数が必要とされる文脈では自動的に実数に変換されます。整数変数に実数値を代入することはできません。実数から整数に変換する方法はあとで説明します。整数型はInteger、実数型はRealという名前です。PurePASCALではReal型は32ビットの単精度のみ用意されています(PurePASCALは実数演算にFLOAT??Xを利用しています。FLOAT2+XとFLOAT3+Xは、それぞれFLOAT2.X、FLOAT3.Xよりも速いのですが、単精度浮動小数点演算にバグを持っており、PurePASCALでの実数演算でおかしな結果が出ることがあります)。

論理型

C言語やBASICでは論理式の値が整数になっていましたが、PASCALには独立した論理型が存在します。論理型は型名Booleanで定義され、TrueかFalseかどちらかの値を取ります。たとえばa=1、b=1のとき、式a=bはTrueで、a=-1、b=1のとき、式a>bはFalseです。

図1 PASCALのデータ型



文字型

文字型の変数には、

```
var c:char;
begin
    :
    c:='A';
    :
end
```

といったようにキャラクタコードが格納されます。Cの文字型は8ビットの整数として使われていましたが、PASCALでは独立した型で、整数との混用はできません。

文字列は後ろでも説明するつもりですが、文字型のPACKED配列として作ります。文字型の定数は'A'のように表現します。漢字などの2バイトコードは文字型には使えません。PurePASCALでは2バイトコードは、文字列の中でだけ使用可能です。

列挙型

列挙型はC言語でも馴染みですね。知らない人のために少し説明します。プログラム中に定数値をマジックナンバーとして埋め込むと、わかりにくくなりがちです。

```
if data=-1 then
と書くよりも、
const ILLEGAL=-1;
と定数を定義しておいて、
if data=ILLEGAL then
```

と書いたほうがよいことはわかりますね。こうすると、あとで仕様変更してILLEGALの値が変わったときでも、最初の定数定義を変更するだけですみます。

この名前付きの定数と似た概念として、列挙型というものがあります。これは、

```
type DAYS=(Mon, Tue, Wed, Thu, Fri, Sat, Sun);
var date:DAYS;
```

のようにすることで、MonからSunまでの値を持つ新たなDAYS型を作ることができるものです。MonからSunまでのそれぞれは、順序数として、0から6までの整数値を持っています。順序数を取り出すには標準関数Ordを使います。名前付きの定数と列挙型の違いは、名前付きの定数というのは単に、定数値に別名をつけたただけなのに対して、列挙型は既存のどの型とも違う新たな型を作り出します。そうすることでプログラムの安全性を高めているのです。論理型Booleanも次のように列挙型の一種と考えることができます。

```
type Boolean=(False, True);
```

よってOrd(False)=0, Ord(True)=1です。

部分範囲型

部分範囲型は、事前に定義された順序型のある範囲の値だけをとる型です。たとえば、

```
type Subrange=1..10;
    Weekday=Mon..Fri;
var i:Subrange;
    d:Weekday;
```

という具合に部分範囲型Subrange, Weekdayを定義すると、Subrange型の変数iは1から10まで、Weekday型の変数dはMonからFriまでの値だけを代入することができるわけです。それ以外の値を代入したら実行時エラーになるでしょう。

このようなチェック機構もPASCAL系の言語の特徴です。バグのために変数に予期せぬ値が代入されるなんてよくある話です。こんなとき、Cだったらデバッグ用のprintfをたくさん入れて再コンパイルということをやっているのではないのでしょうか。ソースコードデバッグがあれば少しはましでしょうが、それにしてもバグの箇所を特定するのにかなり苦勞するでしょう。PASCALだったら多くの場合、ランタイムエラーで一発で見つかるわけです。C言語もこういった機構を取り入れて、

```
int i:1..10;
```

なんて記述ができればいいと思いませんか。チェックはコンパイルスイッチでいつでもoffできるわけですからね。それにいま、shortとかlongとか、ユーザーが決定しているのですが、このように書けたらコンパイラが勝手にintのサイズを決めてくれるわけです。いまCコンパイラを作っている人がいましたら、ぜひ考えてください。規格のあと追っただけじゃ面白くないでしょ。なんて書きましたが、PurePASCALもそのあたりはさぼっていて、部分範囲型も必ず4バイトを占めます。

配列型

配列型の定義は、

type 配列型名=array [添字型] of要素型;
というかたちで行われます。要素型は任意の型、添字型は任意の順序型が指定できます。いくつかの例を挙げると次のようになります。

```
type arrayA=array [-1..10] of Real;
    arrayB=array [Boolean] of array [Char]
of Integer;
    arrayC=array [1..10, 1..10] of Real;
    arrayD=array [DAYS] of Integer;
```

arrayAは添字の下限が-1, 上限が10, 要素が12の実数配列です。arrayBは2×256要素, arrayCは10×10要素の2次元配列です。

arrayCの表記は、

```
type arrayC=array [1..10] of array [1..10]
of Real;
```

のかたちの省略形です。つまり、多次元配列は「配列の配列」と解釈されます。

```
var A:arrayA;
    B1,B2:arrayB;
    C1,C2:arrayC;
    i:Integer;
    date:DAYS;
    E:array [-1..10] of Real;
```

のとき、

```
A [i] :=1.23;
B1 [False] :=B2 [True] ;
C1:=C2;
for date:=Mon to Sun do
    arrayD [date] :=0;
```

といったような操作が可能です。Cとは違って配列の代入が可能で、配列のすべての要素がコピーされます。また、iの値が-1..10のあいだにないときは実行時エラーとなります。

ここでひとつ注意しなければならない点があります。変数Aと変数Eは一見すると同じ型のように見えるのですが、

```
E:=A;
```

のような操作は許されません。実はAとEは別々の型なのです。もっと極端な例を示すと、

```
var va:array [1..10] of Integer;
    vb:array [1..10] of Integer;
```

このとき、vaとvbは別々の型になってしまうのです。もしvaとvbを同じ型にしたいなら、

```
var va, vb:array [1..10] of Integer;
```

または、

```
type array1:array [1..10] of Integer;
var va:array1;
    vb:array1;
```

と書いてください。このように、PASCALの型が同一かどうかの判断は、その構造で判断するのではなく、型の名前かまたは変数が宣言された場所で決まるのです。「面倒でも型には名前をつけろ」ということなのでしょう。これは配列に限ったことではなく、ほかの構造型でも同じです。気をつけましょう。

文字列型

PASCALには文字列のための特別な型は存在しません。Cと同じように、文字列は文字型の配列として表されます。もう少し厳密な定義を示します。

```
type StringN=Packed array [1..N] of Char;
var str:StringN;
```

ただし、 $1 < N$ という約束です。'Packed'を忘れてはいけません。このように定められた文字列型に関しては、同じ文字列型同士の代入、関係演算(inを除く)が許されています。この「同じ文字列型」というのは、簡単にいえばNが同じということです。長さが違う文字列型同士の演算、代入はできません。あまり融通がききませんね。文字列定数は'(single quote)'で囲んで表現します。

```
Const N=25;
type StringN=Packed array [1..N] of Char;
var str1, str2:StringN;
```

```
begin
    str1:='This is character string.';
    str2:='This is string too.'
end;
```

このように、文字列定数の長さが短いときは、帳尻合わせにスペースを入れてください。

集合型

集合型はPASCAL独特のデータ型です。順序型の値の集合をビット列で表現し、集合演算を行うことができます。ただし、PurePASCALではかなりの制限つきで、集合の要素は順序数が0~127のものしか許されません。つまり、集合変数が128ビット=16バイトで表現されるわけです。私自身はあまり利用しないのですが、集合演算が直接行えるのは便利なのかもしれません。集合型の定数は[]の中に、要素を、(カンマ)で区切って並べます。また、要素が連続している場合は途中の要素を全部書く代わりに[3, 10..20, 40]といった記述もできます。

例：

```
type SetOfDays=Set of DAYS;
var weekday, allday:SetOfDays;
begin
    allday:= [Mon..Sun] ;
    weekday:=allday- [Sat,Sun]
end;
```

レコード型

C言語の構造体に当たるもので、いくつかの変数をまとめてひとつの変数として扱うデータ型です。このレコード型と、次に説明するポインタ型を組み合わせると、PASCALは実にさまざまなデータ構造を表現できます。レコード型の定義は、

```
type レコード型名=record固定部  可変部  end;
```

のようになされます。可変部とは、C言語でいう共用体にあたるものです。ありふれた例ですが、複素数型の作り方を例にレコード型の説明をしましょう。


```

type Complex=record
    Re, Im:Real
end;
var a,b,c:Complex;
begin
    c.Re:=a.Re*b.Re-a.Im*b.Im;
    c.Im:=a.Re*b.Im+a.Im*b.Re
end;

```

レコード型Complexは2つのフィールドRe, Imを持っています。ReとImはそれぞれReal型です。Complex型の変数aのフィールドReをアクセスするにはa.Reのように、変数名のあとに.とフィールド名Reを書きます。また、レコード型の変数同士の代入もできます。レコード型の構文規則はかなり複雑なので、構文図をつけておきますから参考にしてください(図2)。

ポインタ型

ポインタ型の変数は対象型と呼ばれる型の変数のアドレスを保持します。C言語のポインタと違って、アドレスの値を整数値に変換したり、あるいはポインタ同士、ポインタと整数のあいだで演算したりすることはできません。また、配列とも関係ありません。ポインタに対して許される演算は、同じポインタ同士の一致と不一致、それからポインタがなにも指していないことを意味する定数Nilとの比較だけです。ポインタ型の定義は、

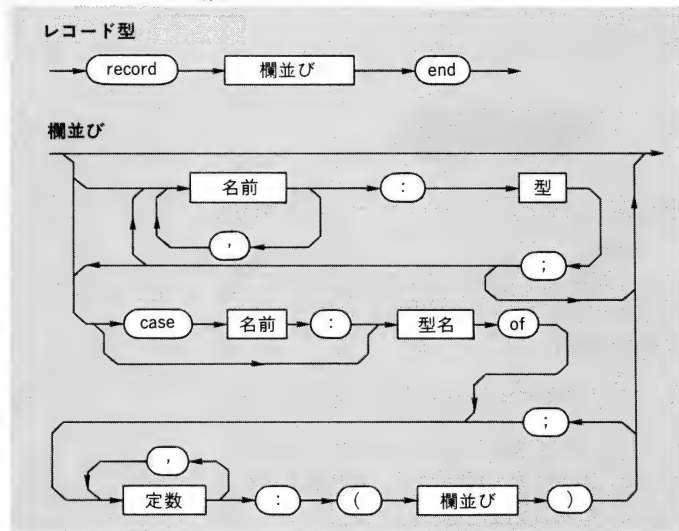
```

type DataPtr= ^Data;
Data=Record
    item:Real;
    next:DataPtr
end;

```

のように、対象型の前に^をつけます。また、対象型にアクセスするときは、ポインタ変数の後ろに^をつけて行います。標準手続きNew(p)は対象型の変数の領域を主記憶上に確保し、そのアドレスをポインタ変数pに格

図2 レコード型の構文



納します。不要になった領域は標準手続きDispose(p)で解放します。図3に双方向リスト、二進木を图示します。これらのデータ構造は次のデータ型Tree, Listで表現できます(よく見たらTreeもListも型の構造は同じですね)。

```

type TreePtr= ^Tree
Tree=Record
    data:Item;
    left, right:TreePtr
end;
ListPtr= ^List
List=Record
    data:Item;
    prev,next : ListPtr
end;

```

ファイル型

ファイル型というのは文字どおりファイルを使うために用意されたデータ型です。ただ、PASCALのファイルの概念は現在のUNIXやMS-DOSのそれとは大きく隔たりがあって、そのままではあまり実用的ではないと思います。詳しくは参考文献を読んでいただくことにして、ここでは概念を示すにとどめます。

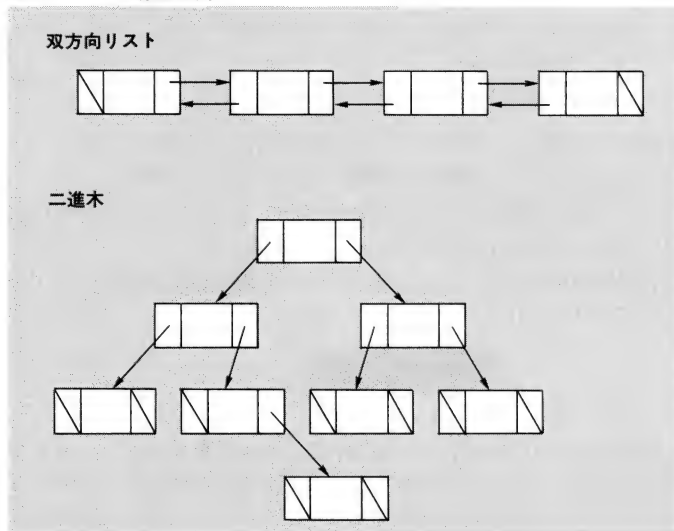
ファイルは次のように宣言されます。

```
var DataFile:File of Data;
```

PASCALのファイルはシーケンシャルファイルで、ファイル処理はいくつかの標準手続きによって行われます。標準手続きを簡単に説明すると、次のようになります。

Reset(f)	ファイルfを読み込みのために初期化する。
Rewrite(f)	ファイルfから要素をひとつ取り、その値をf^へ入れる。
Get(f)	ファイルfから要素をひとつ取り、その値を変数f^へ入れる。

図3 データ構造の例



Put(f) バッファ変数f[^]の値をファイルfに書き込む。

Read(f,x) ファイルfから要素をひとつ取り、その値を変数xに入れる。

Write(f,x) xの値をファイルfに書き込む。

fがファイル変数のとき、f[^] (ポインタみたい)はバッファ変数と呼ばれ、ファイルを読み書きする場合、要素ひとつ分だけのバッファとなります。また、手続きRead, Writeは例外的に不定個の引数を取り、複数の要素を一度に読み書きできます。

```
write(f1,x1,x2,x3);
read(f2,y1,y2,y3);
```

は、

```
write(f1,x1);write(f1,x2);write(f1,x3);
read(f2,y1);read(f2,y2);read(f2,y3);
```

と同じことです。

プログラムの最初に、

```
program main(input, output);
```

と書きますが、このinputとoutputもファイル型の変数です。readとwriteの最初の引数を省略すると、readはinput, writeはoutputが指定されたと解釈されます。また、inputとoutputはテキストファイルと呼ばれる特殊なファイルで、readとwriteはこのテキストファイルに関しては特別な振る舞いをします。詳しくは次の機会に譲ります。

標準関数

PASCALにはいくつかの標準関数が用意されています。PASCALの標準関数はほかの言語と比較すると非常に少ないですが、教育用としては十分でしょう(表1)。それよりも問題となるのは、PASCALの標準関数のいくつかは、PASCAL自身で作ることができないという点です。たとえばPredとSuccは任意の順序型の値を引数として取れることになっていますが、PASCALの言語仕様では引数の型を複数指定することは許されていません。このような汚い点があることをWirthは素直に認めていて、Modula-2では改善したようです。

Chrは一種の変換関数なのですが、C言語で言うところのキャスト演算子と解釈することもできます。事実、MacintoshのTHINK PASCALでは、Cだったら、

(型名) x

と書くところを、

型名 (x)

のかたちで、xの型を“型名”で示される型に変換することができます。整数型から文字型への変換が、ChrじゃなくてCharだったら完璧だったのにね。

実数を引数に取る関数は、整数も引数にできます。なぜなら整数は実数に自動的に変換されるからです。逆に実数→整数の変換は明示的に行う必要があります、そのために2種類の関数(RoundとTrunc)が用意されています。

表1 PASCALの標準関数

関数	引数	戻り値	説明
Abs(x)	実数型または整数型	実数型または整数型(xと同じ)	xの絶対値
ArcTan(x)	実数型	実数型	xの正接
Chr(i)	整数型	文字型	整数型→文字型変換
Cos(x)	実数型	実数型	xの余弦
Eof(f)	ファイル型	論理型	End of File
Eoln(f)	ファイル型	論理型	End of Line
Exp(x)	実数型	実数型	xの指数関数
Ln(x)	実数型	実数型	xの自然対数
Odd(i)	整数型	論理型	iが奇数なら真
Ord(o)	順序型	整数型	順序型→整数型
Pred(o)	順序型	oと同じ順序型	ひとつ前の要素を得る
Round(x)	実数型	整数型	xを四捨五入
Sin(x)	実数型	実数型	xの正弦
Sqr(x)	実数型	実数型	xの自乗
Sqrt(x)	実数型	実数型	xの2乗根
Succ(o)	順序型	oと同じ順序型	次の要素を得る
Trunc(x)	実数型	整数型	xを切り捨て

Round(x)	0 ≤ xのとき、x + 0.5以下の最大の整数
	x < 0のとき、x - 0.5以上の最小の整数
Trunc(x)	0 ≤ xのとき、x以下の最大の整数
	x < 0のとき、x以上の最小の整数

表2 RoundとTrunc

(表2)。

区別が面倒くさいかもしれませんが、0 ≤ xのときは、Roundが四捨五入(丸め)、Truncが切り捨てと覚えておけば十分でしょう。

Ord(o)は順序型の順序数を調べる関数です。oが整数のときは当然o = Ord(o)が成り立ちます。ASCIIコードを採用している処理系ならば(PurePASCALはもちろん) Ord('A') = 65ですね。

また、列挙型の場合、

```
type Colors = (Red, Blue, Green);
```

となっていたら、Ord(Red) = 0, Ord(Blue) = 1, Ord(Green) = 2となるでしょう。PASCALにはこのOrdの逆関数がないのも困りものです。このときPred(Blue) = Red, Succ(Blue) = Greenということになります。Pred(Red)やSucc(Blue)はエラーです。論理型は、

```
type Boolean = (False, True);
```

と考えられるので、Ord(False) = 0, Ord(True) = 1となります。

* * *

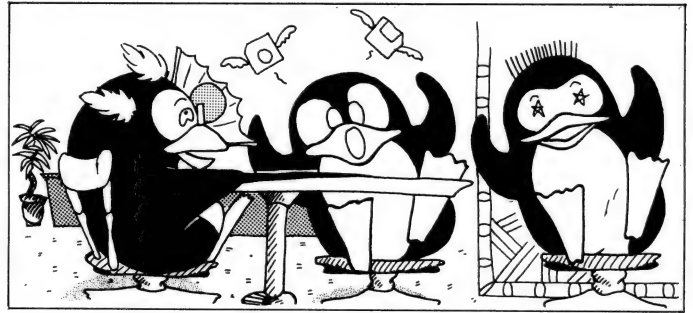
今月はPASCALの持つ豊富なデータ型について説明しました。本当は演算子の説明までやりたかったのですが、型の説明だけでかなり長くなってしまいました。演算子については次の機会に説明することにしましょう。

参考文献

- [1] Niklaus Wirth: "Algorithms + Data Structures = Programs", Prentice-Hall, 1976 (邦訳) 片山卓也: 「アルゴリズム + データ構造 = プログラム」, 日本コンピュータ協会, 1979
- [2] Per Brinch Hansen: "Programming a Personal Computer", Prentice-Hall, 1983 (邦訳) 玄光男: 「パソコンシステムプログラム設計 I. コンパイラ設計編」, 電気書院, 1988

マシン語カクテル in Z80's Bar

第14回——楽な逆ポーランド？——



シナリオ&イラスト：山田純二

特別監修：浦川博之 金子俊一

カラコローン！

マスター（以下M）：いらっしやいませ。

純二（以下純）：どうぼ、ごんばんばあ〜（どうも、こんばんは）。

ようこ（以下Yo）：あら、純二君おひさしぶり。どうしたの？ やばせばびのものまね？

長老（以下老）：それにしてもへたくそじやのう。もっと修業を積まんとお笑い芸人にはなれんぞ。

純：ぶたりども、びどいいがたでずね。ぼぐだって、すぎでやってるんじゃないんでず（2人とも、ひどい言い方ですね。僕だって、好きでやってるんじゃないんです）。

善司（以下善）：趣味でやってんでしょ。

老：同じじや、ばかもの。

純：がぜをびいでしまったんでずよ（かぜをひいてしまったんですよ）。

M：どうせ徹夜でポピュラスでもやっていたんでしょ。

純：おおばたび！ どいだいですが、じつばしがんごんばで、よどおじざわぎまぐって、あざおぎでみだらごうなっていだんでず。（大当たり！ と言いたいです、実は新歓コンパで夜通し騒ぎまくって、朝起きたらこうなっていたんです）。

老：理由はわかったが、そのしゃべり方はなんとかならんか。

純：ぜりぶがにばいばじになってげんごうがずずんでいい、どらいだーばおもっていぬようでずが（セリフが2倍増しになって原稿が進んでいい、とライターは思っているようですが）。

M：そんなことよりツケの残りを早く払ってくださいよ。

純：えっ、なにってんです。先月はプログラムをちゃんと渡したじゃないですか（いきなり元の声に戻った）。

Yo：残念。前回の騒ぎ、全部純二君もち

になっているわよ。

純：そんな、バナナン、ばななん、ば、な、な。

善：空にキラキラお星様……。

Yo：あなたは寝ていなさい。

善：ぐうぐう……。

老：増える増える飲み屋のツケ。大きくなれよ。

純：大きくなってたまりますか。長老が女の子の分はもって、あとはワリカンという話はどうかちやっただんですか？

老：さーて、なんのことかな。わしや知らんぞ。

純：まったく、これだから年寄りば嫌なんだよな。

老：はっはっは。いくら反論してもツケは消えん、かんねんせい。

純：とはほ……。



電卓プロジェクト始動

老：ま、ツケの話はともかく、前回の続きをやってもらおうかの。

M：電卓のプログラムですね。

純：そうです。ちょっとリストが大きかったんで、今月まで残っちゃったんです。

老：数式をちゃんと記述できる電卓を作っておったんじやったな。その式の計算方法はどうかちやっただんですか。

純：それはですね。式をいったん逆ポーランド記法に変換してから計算して答えを求めています。

Yo：ふーん。ポーランド人が逆立ちでもするの？

純：いや、その場合ドンラーポ記法といったほうが……。

老：……逆ポーランド記法というのはじやな。演算子には優先順位があるということを考えてできた計算法のことじやよ。

Yo：優先順位って掛け算と割り算は足し算、引き算よりも先にやるっていうアレのこと？

純：そう。順次処理の好きなコンピュータには逆ポーランド記法が便利なんですよ。

Yo：へえ。なんで？

老：たとえば、 $10 + 5 \times 3$ という式は、まず 5×3 を計算してから10を足すじゃろう。こんなふうにして式の中を先へ進んだり、元に戻ったりするとプログラムがややこしくなってしまうのじや。

純：それが逆ポーランド表記ならさっきの式を例にとってみると、

$10 \quad 5 \quad 3 \quad * \quad +$

と表せるんですよね。

Yo：ただ順番を並べ替えたっただけじゃないの。

老：これにはちゃんとした意味があるのじやよ。純二君や、これを実際に計算することができるかの。

純：もちろん。計算にはスタックを使うんです。左から式の内容を調べて定数はスタックに積んでいき、演算子が見つかったらスタックから定数2つを取り出し計算します。その答えをまたスタックに積み直してこれを終わりまで繰り返せば出来上がり。

老：正解じや。このように逆ポーランド記法を使えば、式の中をいったりきたりせずに左から順番に処理することができるのじやよ。わかったかな、ようこちゃん。

Yo：なるほど、便利にできてるのね。



どうすりゃいいんだ

M：では、逆ポーランド記法への変換のアルゴリズムはどうなっているんですか。

純：まず、式変換ワーク、演算ワークと、このプログラムの心臓部といえる変換テーブル（リストの682～694行参照）を用意し

ます。演算子をどういう順番で書くかはこの変換テーブルが決めてくれるんです。

で、左から順番に式の項を読んでいき、定数は式変換ワークに出力。演算子だったらいま取り出した演算子を横の値、演算ワークのいちばん新しい演算子を縦の値として変換テーブルの内容を取り出します。この内容に従って次のような処理をしていきます。

00 演算ワークのいちばん新しいものを取り出して式変換ワークに出力。取り出した演算子はそのままで同じ処理を繰り返す。

01 取り出した演算子を演算ワークに出力。

02 演算ワークのいちばん新しいものを捨てて、新たに“*”を演算ワークに出力する。

03 演算ワークのいちばん新しいものを、ただ捨てる。

04 演算ワークのいちばん新しいものを捨てて、新たに“*”を2つ演算ワークに出力する。

05 終了

99 エラー

と、どんどん処理を繰り返していくと、式変換ワークに逆ポーランド記法の式が出来る上がるわけです。

Yo: そんなふうにいわれても……。

純: わかりました。それじゃあ、

$10+20+30*40-50$

という式を変換していく様子を見ていきましょう。まず、10は式変換ワークに出力。次は演算子の“+”がくるので変換テーブルの内容を取り出しにいきます。テーブルの縦の値は“+”で、横の値は演算ワークのいちばん新しいもの（この場合はなにも入っていないからワークのエンドコード）。処理内容は01だから、“+”をそのまま演算ワークに出力します。

次の項は20。定数項だから式変換ワークに出力。2番目の演算子“+”のときも1番目と同様にしてテーブルの中身を見る。縦も横も“+”の場合、処理内容は00なので演算ワークのいちばん上の演算子を式変換ワークに出力する。この場合は式から取り出してきた演算子はそのままで、もう一度変換テーブルを見る。今度は縦が“+”，横がワークのエンドコードで処理内容は01となるので演算ワークに“+”を出力する。ここまでで式変換ワークと演算ワークにどのような出力がされているかわかりますか、ようこそさん。

Yo: 式変換ワークには10と20に最初の“+”があって、演算ワークには2番目の

“+”があると思うけど。

純: 正解ですよ、ようこちゃん。これだいたい流れはつかめたでしょう。

老: それで最後には、

$10\ 20\ +\ 30\ 40\ *\ +\ 50\ -$
という式に変換されるわけじゃな。

Yo: なるほどね。



カッコがつくぞ～

M: 純二君、いままでの話を聞いているとわざわざテーブルとか用意なくてもプログラムで何とかかなりそうな気がするけど。

純: 確かにやっていることは優先順位に従って2つの処理を選択しているだけですからね。しかし、この変換テーブルのおかげで括弧を使った式の展開のプログラムがすっきり組めるんですよ。

老: まあ、テーブルを使わずにやってやれないことはないがリストは汚くなってしまいうじゃろうな。

Yo: リストは読みやすいにこしたことはないってことね。

純: じゃ、話も一段落したところで、次はその括弧の話。括弧の使い方には3通りのパターンがあります。

1番目は、 $10+(20+30)$ のように、括弧の中の式をただ優先させるもの。

2番目は、 $10(20+30)$ または $(20+30)10$ のように、前か後ろのどちらかの括弧に乗算が省略されている場合。

3番目は、 $(10+20)(30+40)(50+60)$ のように、前後のカッコに乗算の省略が行われている場合。

プログラムはこのそれぞれの処理に対応させるために3つ用意します。

M: それが変換テーブルの処理番号02, 03, 04ですね。

純: そのとおり。それぞれ1番目には03, 2番目には02, 3番目には04が対応しています。

Yo: ただの括弧と乗算が省略された括弧はどうやって判別するの?

老: それは括弧がどこにでてくるかでわかるのじゃ。数式は定数と演算子が順番に並んでできておるじゃろう。式の解析のときに定数のところで現れたらそれはただの括弧、演算子があるべきところで現れる括弧は乗算が省略されている括弧というわけじゃな。

そして、閉じ括弧のほうは必ず演算子があるはずのところに現れるので、式のもうひとつ後ろの項を調べて判別しなければならない。「演算子、または閉じ括弧」とき

た場合はただの閉じ括弧で、「定数項、または開括弧」ときた場合は乗算が省略された閉じ括弧であると判別できるのじゃ。

Yo: 開括弧、閉じ括弧の両方について調べなくちゃならないのね。

純: 普通だったら乗算の省略は考える必要はないかもしれませんが、やっぱりいつも使っている式をそのまま記述できたほうが気持ちいいですからね。



電卓の使用法

純: じゃあ、最後に電卓の使用法を説明しましょう。

M: あれ、なんだか普通と逆のような気がしますねえ。

老: まあまあ、よいではないか。どちらかといえばアルゴリズムの解説がメインなんじゃから。

純: そういうこと。で、電卓で使えるコマンドは、

? ……メモリの内容を表示

= ……M1～5のメモリに、直前に計算した答えを代入

の2つです。計算はプロンプトに続いて数式を入力すれば、答えが10進と16進で表示されます。数式で使える定数は、10進定数、16進定数（頭に\$を付ける）とメモリのM1～5です。使える演算子は四則演算と余算の“MOD”になっていて、単項演算子や関数はサポートしていません。と、こんなところですよ。

老: 単項演算子や関数もサポートすれば完璧な電卓となったじゃろうに。肝心なところで手を抜きおって。

純: 関数をサポートすると式のネスティングまでやらなくてはなりませんから、それは勘弁してください。

M: というところで、今日はずいぶん頑張って説明しましたね。ごろうさんです。とりえず前回分のツケはこれで払ってもらいましょう。



純：わあーい。

老：じゃあ、そろそろ時間だし、わしは失礼させてもらうか。

純：あ、僕も帰ります。それじゃあ、また

いつか暇になったらやってきます。さようなら。

M：毎度どうも。……と。約1名おいていっちゃったけど、どうしようか。

Yo：そのままでもいいんじゃない。明日はゴミの日だし。

善：ぐうぐう……。

つづく

リスト1

```
9257      357      :DENTAKU in Z80 Bar MAIN
9257      358      :
9257      359      : 1990.5.1 by Junji
9257      360
9257      361      ORG SUBEND
9257      362
9257      363 ENTRY
9257 3E 0C      364 LD A,$0C
9259 CD F4 1F  365 CALL #PRINT
925C CD E2 1F  366 CALL #MPRINT
925F 2A 2A 2A 20 367 DB *** DENTAKU in Z80 Bar ***,$0D
9263 44 45 4E 54
9267 41 4B 55 20
926B 69 6E 20 5A
926F 38 30 27 42
9273 61 72 20 2A
9277 2A 2A 0D
927A 00      368 DB 00
927B      369 MAIN2
927B 3E 3E      370 LD A,">"
927D CD F4 1F  371 CALL #PRINT
9280 11 8E 91  372 LD DE,LIGET
9283 CD D3 1F  373 CALL #GETL
9286 1A      374 LD A,(DE)
9287 FE 1B      375 CP $1B
9289 28 F0      376 JR Z,MAIN2
928B FE 3E      377 CP "-"
928D 20 EC      378 JR NZ,MAIN2
928F 13      379 INC DE
9290 1A      380 LD A,(DE)
9291 B7      381 OR A
9292 28 E7      382 JR Z,MAIN2
9294 FE 51      383 CP "Q"
9296 C8      384 RET Z
9297 21 7B 92  385 LD HL,MAIN2
929A E5      386 PUSH HL
929B ED 73 22 95 387 LD (ERR+1),SP
929F FE 3F      388 CP "-"
92A1 28 07      389 JR Z,WEMPRT
92A3 FE 3D      390 CP "-"
92A5 28 29      391 JR Z,WEMSET
92A7 18 73      392 JR @SHIKI
92A9 C9      393 RET
92AA      394
92AA      395 :MEMORY PRINT
92AA      396 WEMPRT
92AA 11 7B 91  397 LD DE,WEMDAT
92AD 06 05      398 LD B,05
92AF 0E 31      399 LD C,"1"
92B1      400 MEM2
92B1 3E 4D      401 LD A,"M"
92B3 CD F4 1F  402 CALL #PRINT
92B6 79      403 LD A,C
92B7 CD F4 1F  404 CALL #PRINT
92BA 3E 3D      405 LD A,"-"
92BC CD F4 1F  406 CALL #PRINT
92BF 1A      407 LD A,(DE)
92C0 6F      408 LD L,A
92C1 13      409 INC DE
92C2 1A      410 LD A,(DE)
92C3 67      411 LD H,A
92C4 13      412 INC DE
92C5 D5      413 PUSH DE
92C6 C5      414 PUSH BC
92C7 CD 0C 94  415 CALL HXDECPRT
92CA C1      416 POP BC
92CB D1      417 POP DE
92CC 0C      418 INC C
92CD 10 E2      419 DJNZ MEM2
92CF C9      420 RET
92D0      421
92D0      422 :ANSWER TO MEMORY
92D0      423 MEMSET
92D0 13      424 INC DE
92D1 1A      425 LD A,(DE)
92D2 FE 4D      426 CP "M"
92D4 C2 12 95  427 JP NZ,ERROR
92D7 13      428 INC DE
92D8 CD D1 90  429 CALL NUM10
92DB DA 12 95  430 JP C,ERROR
92DE 3E 05      431 LD A,05
92E0 BD      432 CP L
92E1 DA 12 95  433 JP C,ERROR
92E4 7D      434 LD A,L
92E5 B7      435 OR A
92E6 CA 12 95  436 JP Z,ERROR
92E9 7C      437 LD A,H
92EA B7      438 OR A
92EB C2 12 95  439 JP NZ,ERROR
92EE 3E 4D      440 LD A,"M"
92F0 CD F4 1F  441 CALL #PRINT
92F3 7D      442 LD A,L
92F4 C6 30      443 ADD A,"0"
92F6 CD F4 1F  444 CALL #PRINT
92F9 3E 3D      445 LD A,"-"
92FB CD F4 1F  446 CALL #PRINT
92FE 7D      447 LD A,L
92FF 3D      448 DEC A
9300 CD 12 93  449 CALL MEMADR
9303 ED 5B 8C 91 450 LD DE,(ANSWER)
9307 73      451 LD (HL),E
```

```
9308 23      452 INC HL
9309 72      453 LD (HL),D
930A EB      454 EX DE,HL
930B CD 0B 91  455 CALL STRING16
930E CD EE 1F  456 CALL #LTNL
9311 C9      457 RET
9312      458
9312      459 :MEMORY ADDRESS
9312      460 MEMADR
9312 21 7B 91  461 LD HL,WEMDAT
9315      462 MEMR2
9315 B7      463 OR A
9316 C8      464 RET Z
9317 23      465 INC HL
9318 23      466 INC HL
9319 3D      467 DEC A
931A 18 F9      468 JR MEMR2
931C      469
931C      470 :SHIKI NO TENKAI
931C      471 @SHIKI
931C 21 06 92  472 LD HL,WSP
931F 22 86 91  473 LD (WPADR),HL
9322 21 42 92  474 LD HL,ENZSP
9325 22 88 91  475 LD (ESPADR),HL
9328 CD 35 93  476 CALL SHIKI
932B 21 06 92  477 LD HL,WSP
932E 22 86 91  478 LD (WPADR),HL
9331 CD DB 93  479 CALL CALSHIKI
9334 C9      480 RET
9335      481
9335      482 SHIKI
9335 1A      483 LD A,(DE)
9336 FE 00      484 CP $00
9338 20 04      485 JR NZ,SHIKI8
933A      486 SHIKI7
933A 06 0A      487 LD B,$0A
933C 18 47      488 JR SHIKI6
933E      489 SHIKI8
933E FE 29      490 CP "-"
9340 CA 12 95  491 JP Z,ERROR
9343 FE 28      492 CP "("
9345 20 08      493 JR NZ,SHIKI2
9347 13      494 INC DE
9348 3E 06      495 LD A,06
934A CD 2E 90  496 CALL ENZPUSH
934D 18 E6      497 JR SHIKI
934F      498 SHIKI2
934F CD CA 94  499 CALL TEISU
9352 DA 12 95  500 JP C,ERROR
9355      501 SHIKI3
9355 1A      502 LD A,(DE)
9356 B7      503 OR A
9357 28 E1      504 JR Z,SHIKI7
9359 FE 29      505 CP "-"
935B 20 1F      506 JR NZ,SHIKI4
935D 13      507 INC DE
935E 06 09      508 LD B,09
9360 1A      509 LD A,(DE)
9361 FE 00      510 CP $00
9363 28 20      511 JR Z,SHIKI6
9365 FE 28      512 CP "("
9367 28 10      513 JR Z,SHIKI5
9369 FE 29      514 CP "-"
936B 28 18      515 JR Z,SHIKI6
936D 21 62 91  516 LD HL,CALTL
9370 D5      517 PUSH DE
9371 C5      518 PUSH BC
9372 CD 5A 90  519 CALL SEARCHSUB
9375 C1      520 POP BC
9376 D1      521 POP DE
9377 30 0C      522 JR NC,SHIKI6
9379      523 SHIKI5
9379 05      524 DEC B
937A 18 09      525 JR SHIKI6
937C      526 SHIKI4
937C 21 62 91  527 LD HL,CALTL
937F CD 5A 90  528 CALL SEARCHSUB
9382 DA 12 95  529 JP C,ERROR
9385      530 SHIKI6
9385 CD 58 94  531 CALL TBLNUM
9388 B7      532 OR A
9389 28 12      533 JR Z,SPCHK
938B 3D      534 DEC A
938C 28 19      535 JR Z,SPPUSH
938E 3D      536 DEC A
938F 28 1D      537 JR Z,MULTSP
9391 3D      538 DEC A
9392 28 2B      539 JR Z,SPDROP
9394 3D      540 DEC A
9395 28 2E      541 JR Z,DBMULTSP
9397 3D      542 DEC A
9398 28 38      543 JR Z,COMPLETE
939A C3 12 95  544 JP ERROR
939D      545 SPCHK
939D CD 37 90  546 CALL ENZPOP
93A0 C5      547 PUSH BC
93A1 CD 03 90  548 CALL OUTENZ
93A4 C1      549 POP BC
93A5 18 DE      550 JR SHIKI6
93A7      551 SPPUSH
```

```

93A7 78      552      LD      A,B
93A8 CD 2E 90 553      CALL   ENZPUSH
93AB C3 35 93 554      JP      SHIKI
93AE          555      MULTSP
93AE CD 37 90 556      CALL   ENZPOP
93B1 3E 00     557      LD      A,$00
93B3 CD 2E 90 558      CALL   ENZPUSH
93B6 78        559      LD      A,B
93B7 FE 09     560      CP      $09
93B9 CA 55 93 561      JP      Z,SHIKI3
93BC C3 35 93 562      JP      SHIKI
93BF          563      SPDROP
93BF CD 37 90 564      CALL   ENZPOP
93C2 C3 55 93 565      JP      SHIKI3
93C5          566      DBMULTSP
93C5 CD 37 90 567      CALL   ENZPOP
93C8 97        568      SUB     A
93C9 CD 2E 90 569      CALL   ENZPUSH
93CC CD 2E 90 570      CALL   ENZPUSH
93CF C3 35 93 571      JP      SHIKI
93D2          572      :SHIKI END
93D2          573      COMPLETE
93D2 3E FF     574      LD      A,$FF
93D4 01 FF FF 575      LD      BC,$FFF
93D7 CD 40 90 576      CALL   WWRITE
93DA C9        577      RET
93DB          578      :KEISAN MAIN
93DB          579      CALSHIKI
93DB 21 06 92 580      LD      HL,CALSP
93DE 22 8A 91 581      LD      (CALADR),HL
93E1          582      CAL4
93E1 CD 4D 90 583      CALL   WPREAD
93E4 FE FF     584      CP      $FF
93E6 28 05     585      JR      Z,CAL2
93E8 CD 21 94 586      CALL   CALMAIN
93EB 18 F4     587      JR      CAL4
93ED          588      CAL2
93ED ED 73 FA 93 589      LD      (CAL3+1),SP
93F1 ED 7B 8A 91 590      LD      SP,(CALADR)
93F5 E1        591      POP     HL
93F6 22 8C 91 592      LD      (ANSWER),HL
93F9 31 00 00 593      LD      SP,$0000
93FC CD E2 1F 594      CALL   WPRINT
93FF 41 4E 53 57 595      DB      "ANSWER- ",00
9403 45 52 3D 20 596      CALL   HXDECPRT
9407 00        597      RET
9408 CD 0C 94 598      CALL   HXDECPRT
940B C9        599      RET
940C          600      HXDECPRT
940C E5        601      PUSH   HL
940D CD 0B 91 602      CALL   STRING16
9410 3E 28     603      LD      A,"("
9412 CD F4 1F 604      CALL   PRINT
9415 E1        605      POP     HL
9416 CD BE 1F 606      CALL   PRTHL
9419 CD E2 1F 607      CALL   WPRINT
941C 48 29 0D 00 608      DB      "H)",0D,00
9420 C9        609      RET
9421          610      CALMAIN
9421          611      OR      A
9421 B7        612      JR      Z,PTEISU
9422 28 07     613      DEC     A
9424 3D        614      JR      Z,PMEMORY
9425 28 08     615      DEC     A
9427 3D        616      JR      Z,ENZAN
9428 28 0E     617      RET
942A C9        618      PTEISU
942B          619      CALL   PUSHHD
942B CD 0C 90 620      RET
942E C9        621      PMEMORY
942F          622      LD      A,C
942F 79        623      CALL   MEMADR
9430 CD 12 93 624      LD      C,(HL)
9433 4E        625      INC     HL
9434 23        626      LD      B,(HL)
9435 46        627      JR      PTEISU
9436 18 F3     628      ENZAN
9438          629      LD      HL,EZ2
9438 21 52 94 630      PUSH   HL
943B E5        631      LD      H,B
943C 60        632      LD      L,C
943D 69        633      ADD     HL,HL
943E 29        634      LD      DE,JUMPTBL
943F 11 71 91 635      ADD     HL,DE
9442 19        636      LD      C,(HL)
9443 4E        637      INC     HL
9444 23        638      LD      B,(HL)
9445 46        639      PUSH   BC
9446          640      CALL   POPD
9446 C5        641      LD      E,C
9447 CD 1D 90 642      LD      D,B
944A 59        643      CALL   POPD
944B 50        644      LD      L,C
944C CD 1D 90 645      LD      H,B
944F 69        646      RET
9450 60        647      EZ2
9451 C9        648      LD      C,L
9452          649      LD      B,H
9452 4D        650      CALL   PUSHHD
9453 44        651      RET
9454 CD 0C 90 652      :TABLE NUMBER GET
9457 C9        653      TBLNUM
9458          654      PUSH   DE
9458 D5        655      LD      HL,ENZTBL
9459 21 72 94 656

```

```

945C C5        661      PUSH   BC
945D 78        662      LD      A,B
945E          663      OR      A
945E B7        664      LD      DE,08
945F 11 08 00 665      JR      Z,TBNUM2
9462 28 03     666      TBNUM3
9464          667      ADD     HL,DE
9464 19        668      DJNZ   TBNUM3
9465 10 FD     669      TBNUM2
9467          670      POP     BC
9467 C1        671      PUSH   HL
9468 E5        672      LD      HL,(ESPADR)
9469          673      LD      E,(HL)
9469 2A 88 91 674      POP     HL
946C 5E        675      ADD     HL,DE
946D E1        676      LD      A,(HL)
946E 19        677      POP     DE
946F 7E        678      RET
9470 D1        679      ENZTBL
9471 C9        680      DB      " / MOD + " * ( SN
9472          681      DB      "00,00,01,01,01,01,01,01 :
9472 00 00 00 01 682      DB      "00,00,00,01,01,01,01,01 :
9476 01 01 01 01 683      DB      "00,00,00,01,01,01,01,01 :
947A 00 00 00 01 684      DB      "00,00,00,01,01,01,01,01 :
947E 01 01 01 01 685      DB      "00,00,00,01,01,01,01,01 :
9482 00 00 00 01 686      DB      "00,00,00,00,01,01,01,01 :
9486 01 01 01 01 687      DB      "00,00,00,00,01,01,01,01 :
948A 00 00 00 00 688      DB      "01,01,01,01,01,01,01,01 :
948E 01 01 01 01 689      DB      "01,01,01,01,01,01,01,01 :
94A2 01 01 01 01 690      DB      "01,01,01,01,01,01,01,01 :
94A6 01 01 01 01 691      DB      "00,00,00,00,04,02,99 :
94AA 01 01 01 01 692      DB      "00,00,00,00,02,03,99 :
94AE 01 01 01 01 693      DB      "00,00,00,00,99,99,05 :
94B2 00 00 00 00 694      :TEISU NO HANTEI
94B6 00 04 02 63 695      TEISU
94BA 00 00 00 00 696      CALL   TEICHK
94BE 00 02 03 63 697      RET     C
94C2 00 00 00 00 698      LD      A,(DE)
94C6 00 63 63 05 699      CP      "M"
94CA          700      JR      Z,TEISU2
94CD CD 04 95 701      CP      "S"
94CE 1A        702      JR      Z,HEXTE1
94CF FE 4D     703      CALL   NUM10
94D1 28 0D     704      RET     C
94D3 FE 24     705      DECTEI
94D5 28 1B     706      SUB     A
94D7 CD D1 90 707      LD      C,L
94DA D8        708      LD      B,H
94DB          709      JR      TEISU3
94DB 97        710      LD      A,B,A
94DC 4D        711      INC     DE
94DD 44        712      LD      A,(DE)
94DE 18 0E     713      SUB     "1"
94E0          714      RET     C
94E0 47        715      CP      04
94E1 13        716      CCF
94E2 1A        717      RET     C
94E3 D6 31     718      INC     DE
94E5 D8        719      PUSH   DE
94E6 FE 04     720      CALL   #HLHX
94E8 3F        721      POP     BC
94E9 D8        722      JR      NC,DECTEI
94EA 13        723      LD      E,C
94EB 4F        724      LD      D,B
94EC 3E 01     725      CALL   #2HEX
94EE          726      LD      H,00
94EE CD 40 90 727      LD      L,A
94F1 C9        728      JR      DECTEI
94F2          729      INC     DE
94F2 13        730      PUSH   DE
94F3 D5        731      CALL   #HLHX
94F4 CD B2 1F 732      POP     BC
94F7 C1        733      JR      NC,DECTEI
94F8 30 E1     734      LD      E,C
94FA 59        735      LD      D,B
94FB 50        736      CALL   #2HEX
94FC CD B5 1F 737      LD      H,00
94FF 26 00     738      LD      L,A
9501 6F        739      JR      DECTEI
9502 18 D7     740      LD      A,(DE)
9504          741      CP      "M"
9504 1A        742      RET     Z
9505 FE 4D     743      CP      "S"
9507 C8        744      RET     Z
9508 FE 24     745      CP      "0"
950A C8        746      RET     C
950B FE 30     747      CP      "9" -> 1
950D D8        748      CCF
950E FE 3A     749      RET
9510 3F        750      :ERROR PRINT
9511 C9        751      ERROR
9512          752      CALL   #LTNL
9512          753      CALL   WPRINT
9512 CD EE 1F 754      DB      "ERROR!!",0D
9515 CD E2 1F 755      DB      "00
9518 45 52 52 4F 756      LD      SP,$0000
951C 52 21 21 0D 757      ERR
9520 00        758      RET
9521 31 00 00 759
9524 C9

```


祝! 1周年記念

Komura Satoshi 古村 聡

今回こそは本当に1周年記念だよー。げに恐ろしきは勘違いかな。さて、今回のショートプロはどこかで見たことがあるようなX1用ゲーム「THE FANFAN」とちょっと変わったX68000用「かべくすし」です。おまけの企画もあるよ。



illustration : T.Takahashi



1周年のごあいさつ

どーもっ！ いきなり原稿が落ちてしまったり、なぜか1周年の前夜祭を開いてしまったりといろいろアクシデントもありました。が、ついにこのショートプロはーいも本当の1周年を迎えることができました。めでたいめでたい！ ということで特別企画として囲みを用意しましたのでぜひ読んでくださいわ。

いやあ、それにしてもこのショートプロの企画が出たときは「とりあえず3カ月がんばってね」ということだったんで、まさかショートプロ1周年、さらにハンズ延長戦突入（ハンズを読んでね）というところまで続くとは思ってもなかったんですよ。これもひとえに、いつも楽しいイラストを描いてくれる高橋哲史くん、毎月のように破られる締め切りに「おい、明日は原稿持ってくるんやろな」とドスのきいた関西弁で励ましてくれる編集担当様、いつもひとをオモチャにして遊んでくれるスタッフのみんな、そしてやっぱり、プログラムやらハガキやら毒物飲料40本入りの段ボール箱やらでいろいろと連載にネタを提供してくれる読者の皆さんのおかげなのです。本当に本当にありがとうございます。これからも私のこと



なんでいろいろとアクシデントもあろうかとは思いますが、これからも見捨てないでショートプロバ一ていを読んでやってください。よろしくお願いします。

以上、(て)からの1周年のごあいさつで
した。



ピポパで勝負!

さーと、そろそろいつものショートプロをいきますか。今月の1本目は「自信があります」(おおっ)と言い切ってくれた遠藤さんの反射型アクションゲーム「THE FANFAN」です。

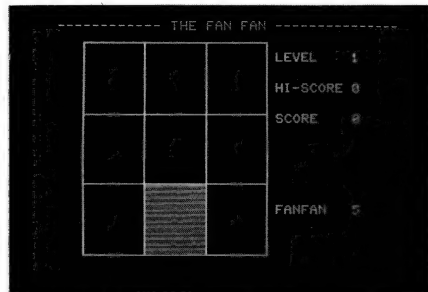
THE FANFAN for X1 シリーズ

(CZ-8FB01)

栃木県 遠藤亮司

プログラムをRUNさせると画面にマス目
が描かれます。実はこのマス目はテンキー
の1から9までのキーに対応しているんです。
スペースキーを押すとマスが「ぱっ、
ぱっ」と赤く点滅していきますから、しっ
かりとその順番を覚えておいてください。
あ、動き終わりましたね。次はプレイヤー
の番です。ピポパとマスが赤く光ったのと
同じ順番で対応する1から9のキーを押して
やってください。ピポパ。点滅したのと
同じ順番でうまくテンキーを押すことがで
きれば1面クリア。とーぜんですけど、面
が進むごとに(2面ごとにだけどね)「ピポ
パ」の数が増えて、だんだん難しくなっ
ていきます。テンキーを押すのを5回失敗し
てしまうとゲームオーバーです。

……でえーい、ちゃかちゃか、ちゃか
 ちゃか、動くんじゃねえやい。覚え切れない
 じゃないか！ 結構なスピードでマスが点
 滅していくので順番をちっとも憶えられな
 いんですよー。うーん、なかなか瞬間
 的な記憶力を要するゲームです。こりゃー
 確かに反射型アクションゲームだ。いや、
 正確には「反射的な記憶力をつけるゲーム」



THE FANFAN

かもしれない。なんかこのゲームをやったあとして爽快な感じになれますね。もっとも、この手のゲームってパズルゲームの次に神経衰弱が苦手な私としては（苦手なもの多い男だな）結構つらいゲームでもあったりするんですけどね。

ん、なにに、「仮面ノリダーのファンファン大佐が使っていたモニタディスプレイからこのゲームを思いつきました。よって、THE FANFANであります」ってか。なるほどねー。ふとゲームのネタを思いついたというのはよく聞く話ですけど、テレビからネタを拾ってくるというのは結構いい手かもしれないですね。テレビにしてもゲームにしても人を楽しませるものであることには変わらないわけですから。やっぱり人を楽しませようと思ったら、エンターテイメントの先輩であるテレビやマンガを見習うというのは結構いい方法じゃないかなーと思ったりします。

もっとも、安易にテレビからキャラクターを借りてきてクソゲーなんか作っちゃうと大ひんしゆくを買いかねないわけだけど(しかし、これもよくある話)。



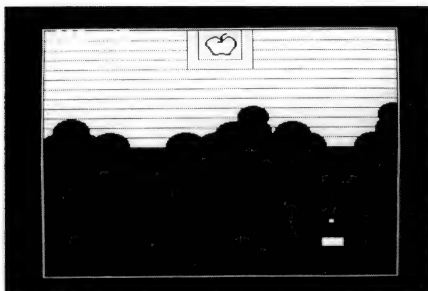
縮めて縮めたかべくずし

さて、続いて今月の2本目です。

かべくずし for X68000

(X-BASIC)

東京都 太田敬三



かべくずし

パドルを動かしてタマに当てて、画面上半分の壁をくずし、画面中央の絵を破壊する。そう、早い話がブロックくずしなんです。なーんだと思ったそのあなた、あなたは甘い、甘すぎるっ！ つい力が入ってしまいましたが、そこはショートプロに載るプログラム、当然ただのブロックくずしではありません。

まず、パドルはジョイスティックで8方向に動かすことができます。そう、左右だけに動くのではないんです。んで、さらにジョイスティックのボタンを押しながらだと速く動いて、そーれそーしたタマと追いかけてこた、てなもんです(ちなみにボタンを2つとも押しちゃうと、ほとんど操作不能くらい速くなる)。ほらねっ、普通のとはちょっとばかし違うでしょ。

うーん、それはともかく、タマが壁にぶつかったときがいいですねえ。普通みたいには当たった1ブロックが消えるんじゃなくて「パバパッ」と爆発して、まるーくえぐれるんですね。そして、ブロックくずしのように玉がブロックとブロックの間で往復運動したりもする(この表現でわかるかな?)わけなんです、このときなんか爆発してるのがとーっても綺麗です。そうそう、ターボボタン(ジョイスティックのトリガーを押すとパドルの移動が速くなる)機能は私が勝手につけてしまいました。作者の太田さん、ごめんなさい。

ちなみにどうやったかという、パドルの移動ルーチン(360,370行)のところで、strig(1)関数で取ってきた値をパドルの移動の増分値にかけてるだけです(だから、ボタンの左右どちらが押されたかで速さがちがう。そのうえ、同時に押すと上記のようにメチャ速くなるわけ)。

いやそれにしても、ゲーム自体もたいしたものだけど、それ以上によくプログラムを小さくまとめたなーと感心してしまいました。特に、破壊目標である絵。よくこの小さいプログラムでこんな絵を表示させましたねー。ええ、この短いプログラムのどこにそんな絵のパターン(しかも毎回毎回

リスト1 THE FANFAN

```

10 CLS4:WIDTH 40:INIT:DIMA(7,7),B(7,7):PRW 254:HI=0:REPEAT OFF
20 LINE(24,16)-(192,183),PSET,B
30 LINE(80,16)-(136,183),PSET,B
40 LINE(24,72)-(192,127),PSET,B
50 FOR I=1 TO 7:COLOR2:PRINT " ":NEXT
60 GET@(0,0)-(7,7),A:CLS
70 GET@(0,0)-(7,7),B
80 COLOR4:PRINT "----- THE FAN FAN -----":COLOR5
90 N=3:F=0:F1=1:LE=1:SC=0:OV=5
100 GOSUB540
110 LOCATE 25,21:PRINT"HIT SPACE KEY"
120 IF INKEY$=" " THEN130 ELSE120
130 LOCATE 25,21:PRINT " "
140 FOR L=0 TO N
150 XX=INT(RND*3):YY=INT(RND*3)
160 IF XX=0 THEN X=3
170 IF XX=1 THEN X=10
180 IF XX=2 THEN X=17
190 IF YY=0 THEN Y=2
200 IF YY=1 THEN Y=9
210 IF YY=2 THEN Y=16
220 PUT@(X,Y)-(X+7,Y+7),A:PLAY"L0GFG"
230 PUT@(X,Y)-(X+7,Y+7),B
240 IF F=1 THEN RETURN
250 IF X=3 ANDY=2 THEN K=7
260 IF X=3 ANDY=9 THEN K=4
270 IF X=3 ANDY=16 THEN K=1
280 IF X=10ANDY=2 THEN K=8
290 IF X=10ANDY=9 THEN K=5
300 IF X=10ANDY=16 THEN K=2
310 IF X=17ANDY=2 THEN K=9
320 IF X=17ANDY=9 THEN K=6
330 IF X=17ANDY=16 THEN K=3
340 POKE &HC000+L,K
350 NEXT
360 FOR L=0 TO N
370 K$=INPUT$(1)
380 K1=PEEK(&HC000+L):F=1
390 IF K$="1" THEN X=3 :Y=16:K2=1
400 IF K$="2" THEN X=10:Y=16:K2=2
410 IF K$="3" THEN X=17:Y=16:K2=3
420 IF K$="4" THEN X=3 :Y=9 :K2=4
430 IF K$="5" THEN X=10:Y=9 :K2=5
440 IF K$="6" THEN X=17:Y=9 :K2=6
450 IF K$="7" THEN X=3 :Y=2 :K2=7
460 IF K$="8" THEN X=10:Y=2 :K2=8
470 IF K$="9" THEN X=17:Y=2 :K2=9
480 IF K2=K1 THENSC=SC+5:GOSUB220 ELSE PLAY"DCD":L=L-1:OV=OV-1
490 GOSUB540
500 IF OV=0 THEN600
510 NEXT:F=0:F1=F1+1
520 IF F1>2 THEN LE=LE+1:N=N+1:F1=1
530 GOTO100
540 IF HI<SC THEN HI=SC
550 LOCATE25,3:PRINT"LEVEL ";LE
560 LOCATE25,6:PRINT"HI-SCORE";HI
570 LOCATE25,9:PRINT"SCORE ";SC
580 LOCATE25,18:PRINT"FANFAN ";OV
590 RETURN
600 LOCATE25,21:PRINT"++GAME OVER++"
610 FOR T=0 TO 5000:NEXT:CLS:GOTO80

```

リスト2 かべくずし

```

10 int a,b,ch,ta,k,x,y,m,n,v,w,i,j,sc,sa,hs,sg:dim char z(255)
20 color 3:screen 0,1,1,1:apage(1):box(0,0,255,255,1):apage(0)
30 sp_init():sp_disp(1):sp_on(0,1):m_alloc(1,10)
40 for a=0 to 2:for b=0 to 2:z(a*16+b)=14-(b=1):next:next:sp_def(1,z)
50 for a=0 to 7:for b=0 to 14:z(a*16+b)=14-(a=2)-(a=3):next:next:sp_def(0,z)
60 randomize(543*atoi(right$(times$,2))):ch=int(rnd()*19)+165
70 while 1
80 color 7:sc=0:ta=0
90 while 1
100 start():if game() then break
110 for a=0 to 9999:next
120 ta=ta+20:if ta>120 then sc=sc+50000:ta=0
130 endwhile
140 for a=0 to 127:box(a,a,255-a,255-a,0):next
150 locate 0,1
160 if sc>hs then hs=sc:print"こりゃあすごいなハイスコアだよ!" else (
170 print"めざすは";hs;"がんばろう!")
180 color 3:repeat:until strig(1)
190 endwhile
200 end
210 func start()
220 sa=10:v=int(rnd()*224)+16:w=248:i=0:j=0:x=v:y=246:m=4*int(rnd()*2)-2:n=-3
230 ch=ch+1:if ch=184 then ch=165
240 wipe():cls:locate 0,0:print sc
250 a=int(rnd()*192)

```


絵が違う)があるんだ? と、思わず探してしまっただけじゃないですか。ふーん、そうか。X68000にはこういう絵のパターンがちゃんとあって、そうやると絵が出せるんですか。私は知りませんでした。ねえねえ、太田さん、どこでこんな方法を知ったのかこっそり教えてくださいませんか。(←外字定義をしたことのないやつ)

ただ、このゲームよくできているんだけど、壁に当たったときパドルの動きが止まっちゃうのと、あとリスト中に全然注釈がないのがちょっと残念なんだよねー。これから投稿する人はぜひプログラムに注釈をつけてくださいね。

うーむ、それにしてもこのコーナーはいつまで続けられるのかなー。とりあえずマシン語カクテルとはスタートがほとんど同じなので負けたくないな。そんなこんなでまた来月。

```
260 for b=2 to 15:palet(b,hsv(a,31,31)):a=a+4:a=a+192*(a>191):next
270 a=a+66:a=a+192*(a>191):palet(1,hsv(a,31,31))
280 for a=0 to 11:fill(2,a*10+2+ta,253,a*10+10+ta,a+3):next
290 box(104,1,151,41,0):fill(105,2,150,40,2)
300 box(112,1,143,31,0):fill(113,2,142,30,15)
310 symbol(116,3,chr$(235)+chr$(ch),1,1,2,0,0)
320 sp_move(0,v-7,w,0):sp_move(1,x-1,y-1,1):repeat:until stick(1)
330 endfunc
340 func game()
350 k=stick(1):sg=strig(1)
360 i=(3+sg*3)*((k=1)+(k=4)+(k=7)-(k=3)-(k=6)-(k=9)):v=v+i
370 if v and 256 then v=v-i
380 j=(k/6)-(k/4)+(k=0) shl (2+sg):w=w+j:if not w and 128 then w=w-j
390 x=x+m:if x and 256 then m=-m:x=x+m
400 if point(x,y) then m=-m:if dokan(point(x,y)) then return(0)
410 y=y+n:if y and 256 then if y<0 then n=3:y=0 else return(1)
420 if point(x,y) then n=-n:if dokan(point(x,y)) then return(0)
430 sp_move(1,x-1,y-1,1):sp_move(0,v-7,w,0)
440 if abs(v-x)<9 and abs(w-y)<4 then pakon()
450 goto 350
460 endfunc
470 func dokan(a)
480 for b=1 to a:circle(x,y,b,1):next
490 m_init()
500 if a=15 then sa=sa+11111:m_trk(1,"V15@6801C") else m_trk(1,"V10@6802C")
510 m_play():sc=sc+sa:locate 0,0:print sc;sa;chr$(5):sa=sa+11
520 for b=1 to a:circle(x,y+1,b,0):circle(x,y,b,0):next
530 return(a=15)
540 endfunc
550 func pakon()
560 if n<0 then return()
570 m_init():m_trk(1,"V10@6706E"):m_play()
580 sa=10:n=-3:if abs(i+m)=1 then m=-m
590 endfunc
```

1周年特別企画——どんちゃん騒ぎの部屋

えー、ささやかながら1周年特別企画として(手前ミソくさくてちょっと恥づかしいんだけど)皆様からショートプロばーていに寄せられたご意見、ご感想、文句に苦情、祝辞の言葉などなどにお答えしたいと思います。最初の方、どーぞ!

☆ほう。古村氏の連載も1周年ですか。月日は百代の過客にして天上天下唯我独尊。で、人間には3種類あると仮定しよう。(1)普段は無口だが、文章を書かせるととても面白くて含蓄深いことを書く奴、(2)普段のノリがそのまま文章に出る(それ以外は書けなかったりする)ヤツ、(3)普段は面白いのに、文章はまったくつまらないやつ、である。(で)君はといえば、いわずもがな(2)である。彼はあのとおり的人格なのだから。というわけで、変にウケ狙いなどせず、すくすく伸びて、成長した姿を読ませてほしい。それが非常に楽しみである(爆笑)。(荻窪圭)

へへーっ! 荻窪師匠からの祝辞だー! いつもお世話になってます。そう、私はそれ以外は書けないんです。ちなみに荻窪師匠は(4)書いている文章もすごいが本物に会ってみるとさらにすごいので恐れ入ってしまうタイプ。つまり人間がはるかに深い(人物から文章の想像はつくけど文章から人物を想像できない)っていうことで尊敬しています。はい。今度飲みにいきましょうよ、師匠。もちろん荻窪師匠のおごりでね! しかし、悪友金子俊一とかグラフィックの魔術師丹明彦さんにも祝辞を頼んだのになー。いったいどーなってるんだろ。☆(で)さんの初登場は(ビー)年(ビー)月号の(ビー)のレビューではないですか? 違っていたらすいません。

(アンケートハガキより、原正人さん)ピンポンピンポン! 大正解です。えー、あの頃は(で)って使ってなかったのによくわかりましたねー。まだ、Oh!X編集部がいまの泉岳寺に移るまえのまえ、四番町の半地下の編集部の頃の話だからねー。懐かしいなあ。ちなみに

本文中の「ビー」は私がつけたものですが、別に恐ろしいことが書いてあるわけではありません。あしからず。

☆5月号の「空飛ぶDNAデモ」を走らせてみた。それを見た友人曰く、「まんが日本昔ばなし」のオープニングみたいだと。

(アンケートハガキより、神生直敏さん)

おー。「ほうやー、い子だ〜♪」というあれですね(そういえばパロディで「ほうやーい子だ金だしな」というのがあったな)。しかし、あのデモは本当に好評でした。そうそう、某MS-DOSマシンにも似たようなデモがあるという話を聞いたのである人に見せてもらったのですが、見た瞬間、「勝った!」と思ってしまいました。

☆ちょーどゲームでも作ってみようと思ったとこなんです、(で)さん。シューティングじゃないけど。4月号の外部関数は役に立ちそうです。(アンケートハガキより、小林到さん)

わーい、それはよかった。うれしいです。ばーていハンズはなかなか評判がよいので喜んでおります。それに4月号のsp_chk()も5月号のデモに負けず劣らず好評でした。ここんとこいい投稿が多い。ゆえにショートプロの評判も上がるというわけでとてもうれしい。小林さんもぜひ投稿してみてくださいね。

☆ライターのプロフィールが知りたい。

(アンケートハガキより、桐山秀幸さん)

ショートプロとは関係ないけど、思わず持ってきてしまいました。あははは、私も知りたい。うちの編集部は謎の人物がいっぱいいるから壮絶なものになること間違いなし。でも、自分のプロフィールは遠慮したい……。

☆すいませーん。Reserved featureエラーがでるんですけどー。(バグ電話より)

すいませーん。X1のBASIC(CZ-8FB01)には新旧のBASIC(ver.1と2)があるのはご存じです。ショートプロのものはほとんどがどちらのBASICでも使えるのですが、5月号のDIG

MANは旧BASIC専用だったんですよー。うっかり私が書き忘れてたんです。本当にごめんなさい。今後のために(やらないように心がけるつもりではありますけど)一応、こういうときの対処の方法を覚えておきましょうね。とりあえず、バージョンの違うBASICで打ってしまったらASCIIセーブしてください。

SAVE「ファイル名」, A

それからリセットして本来使うはずだったBASICを立ち上げます。そして、再びロードすればOKです。

おお、そうだ。ショートプロで質問の多かったものに5月号のDNAデモがあるんですが、これはコンパイル時のスイッチを小文字にしていた人が多かったみたいです。コンパイルできなかった人はそこを注意してもう一度やってみてください。リストにバグはありません。

うーむ、なにやら「あの筋?」質問箱になってしまった。

☆(で)のばーていハンズ(その3)はものすごくうれしい。

(アンケートハガキ、白井達広さん)

ありがとー。私も本当にうれしいです。あのハンズって結構大変なんですよ。なにしろOh!Xには珍しく毎月ちょっとずつプログラムを載せていく形式なんで始める前の下準備がめんどくさいわ、文字が小さいから1ページでショートプロ3ページ分の文章を書かなくちゃいけないわなんですよ。その努力が報われたわけで、いや、よかったよかった。延長戦もよろしくね(あと、リクエストもね)。

うーん、アンケートハガキっていいなあ。と思つてるとこんなハガキもあつたりします。

☆(で)のばーていハンズのコーナーを3ページくらいに増やしてほしい。

(アンケートハガキ、箕浦健一郎さん)

……かんべんしてよ(で)もうれしい!。ま、なにはともあれ、これからもよろしくお祈りします。

恵まれている(で)に愛の手を!

さて、さてさて。結構のんびりやっていたはずのこのコーナーもいよいよ今月の敵と敵のタマの動き、そして来月の当たり判定を残すのみにってしまったんです。ということで本来なら来月で「それではみなさん、さよならー」となるはずだったのですが、皆様のハガキのおかげで再来月からぱーていハンスは第2部に突入することとなりました。はい、拍手拍手! でも、連載が延長になるのはうれいしんだけど、まさかこうなるとは予想すらしてなかったんで、はつきりいつてまだなにをどうするのか全然準備してなかったりするのですよねー。困ったなー、急になんか作れていわれてもなに作っていいのかわからんよー。てなわけでこんなものを作ってほしいとかこうしてみてはどうかとか、こういうところがわからなかったとかいうハガキを大募集しちゃいます。ネタのない(で)に、あいの手をー! あーこりゃこりゃ(そのあいの手じゃないー)。

敵襲だーっ! ゲームの個性だーっ!

さて。というわけで、敵の出現、敵の移動、敵のタマ撃ち、敵のタマの移動です。

シューティングってゲームセンターにもいろいろなものがありますが、基本的には自機を動かしてタマを撃つという意味でそんなに変わらないですね。シューティングゲームの個性って敵の出現、動きのパターンや背景なんかかなりの部分を占めていると思うんです(例外も多々ありますが)。だからシューティングゲームを作るとき、背景をカラフルにしたり、デカキャラを作ったり(X68000だったら簡単でしょ)、敵キャラの動きをなめらかにしたり、あと、敵のタマが多くなりすぎてバランスが悪くなってしまわないようにとかの努力をすれば市販ゲームぐらいにできなくはないと思うんですよ。特に、X68000みたいにスプライトやBGがあったりなどなどという機能が揃ってるマシンだとアフターバーナーみたいに特別なプログラムテクニクが必要なものでない限り、アマチュアの作ったゲームと売ってるゲームの差は、極端にいうとどれだけデータを作れるか(どれだけの人か)とどれだけ時間をかけたか、どれだけ妥協しないで作ったかの差ではないかだと思います。ゲーム作りの極意は根性(もちろん創意工夫も)なんです。決してテクニクだけではありません。

皆さんにはそのようにがんばっていただきたいなー、ということで今回私は手を抜かせていただきます(い、いままで並べたゴタクはいったいなんだっただけ……)。

で、敵の動きなんですがとりあえずこんなのを考えてみました。

「敵がすすーっと下りてくる」

「ばっ、とタマをまき散らす」

「敵はすすーっと逃げていく」

なんかとんでもなくいやな性格してる敵キャラですけどねー。んで、敵をとりあえず動かす

てみたいんですけど、その前にちょっと思い出さなきゃいけないことがある。そうそう、先月いったあれなんです。自機も敵も同時に動かさなくちゃいけないので、かわりばんこで動かすように組んでやらなくちゃいけないんですよね。先月、自機とタマを交互に動かすために自機のメインルーチンの中に、

```
firemove()
```

って1行入れて自分のタマを動かすルーチンと呼び出しましたよね。それと同じように、

```
enemy_move()
```

って1行入れて敵を動かすルーチンと呼び出してやるんです。ちなみに敵のタマを動かしてやるルーチンが、

```
bomb_move()
```

なんですけど、敵のタマもちろん同時に動くわけですよ。だからbomb_move()もそこに入れていい……んですが、なぜかbomb_moveはenemy_moveが呼び出しています。別にこれは意味はないんです、っていうか実はなんでこうしたのかよく憶えてないんです(こらこら)。たぶん敵が動くルーチンと敵のタマを動かすルーチンだけほかのルーチンと別の日に作ったので思わずそうしてしまっただけじゃないかな。別に次々とルーチンがルーチンを呼んでもかまわないっちゃかまわないんですが、やっぱりリストが読みにくいのでみなさんはちゃんとどこからかに統一しましょうね。

それはそうと敵が出てきて引き返す(折り返す)っていうほうがわかりやすいかな? ってことは、まず、敵がどこで折り返すか決めておいて、それから敵をつつと下ろして行って、折り返し位置にきたら帰っていくようにすればいいわけですね。さて、ここで問題です。ここではいくつ変数を作ればいいでしょう。

自分のX座標、およびY座標

折り返し点のY座標

自分が上がっているか下がっているかのフ

ラグ

うん、4つもあればよさそうですね。自分が上がっているか下がっているかのフラグはたとえ

上がっているとき=-1

下がっているとき=+1

としてやれば敵を動かすときに(たとえば敵のY座標がenemy_y、フラグがenemy_sgnという名前だとしたら)、

```
enemy_y = enemy_y + enemy_sgn
```

としてやればできそうですね。

それじゃ、1つひとつルーチンを作っていきますか。まずは敵の出現。

```
• enemy_appear()
```

とりあえず、

「タマは出ていないか」

「自分のX座標と引き返し座標を決める」

「上がり下がりフラグを+1にする」

このくらいかな。で、これを敵のY座標が0の(つまり敵が現れていない)とき、このルーチンと呼んでやればいいわけね。んで、

```
• enemy_move()
```

出てきた敵をこのルーチンで動かす。これは敵を順番に1ステップ動かすわけですね。んで折り返し点にきたら上がり下がりをして1にしてタマを出させます。

```
• bomb_move()
```

タマが出ていたらタマを1ステップ進める。で、「タマをばらまく」ことにしたわけですが、とりあえずタマは3つ出して左下、下、右下に進めます。

あー疲れた。とりあえずこんなもんかなー。

さて、来月は当たりチェックやおしまいね。んー、でも当たりチェックだけで1ページもたすの苦しうだなー(たぶん1/4ページくらいで終わっちゃうと思うんだよねー)。ま、いいか。明日は明日の風が吹くと。来月またこのOh! Xで。ガガガガ(と穴を掘って去る)。

```
330 enemy_move()
330 /* 敵の動き */
340 func enemy_move()
350 for i=0 to 2
360   if enemy_y(i)>enemy_b(i) then enemy_sgn(i)=-1:enemy_fire(i)
370   if enemy_y(i)>0 then enemy_y(i)=enemy_y(i)+enemy_sgn(i)*8
380   j=0:for a=0 to 2:j=j+bomb_y(i,a):next
390   if j=0 and enemy_y(i)=0 then enemy_appear(i)
400   sp_set(38+i,enemy_x(i),enemy_y(i),(enemy_sgn(i)-1)*2)*&H8000*&H123)
410 next
420 bomb_move()
430 endfunc
440 func enemy_appear()
450 enemy_b(i)=0:while (enemy_b(i)=0 or enemy_x(i)<16 or enemy_x(i)>193)
460   enemy_x(i)=rand()and&HF0:enemy_y(i)=8:enemy_sgn(i)=1
470   enemy_b(i)=rand() and &HF0
480 endwhile
490 endfunc
500 func enemy_fire():/*敵もタマを撃つ */
510 for j=0 to 2
520   bomb_x(i,j)=enemy_x(i):bomb_y(i,j)=enemy_y(i)
530 next
540 endfunc
550 func bomb_move() /*タマの動き*/
560 for i=0 to 2
570   for j=0 to 2
580     if bomb_y(i,j)>0 then bm_sub(i)
590     next
600     next
610     next
620 func bm_sub()
630   bomb_y(i,j)=bomb_y(i,j)+8
640   if j=0 then bomb_x(i,j)=bomb_x(i,j)-8:if bomb_x(i,j)<=0 or bomb_y(i,j)>256
650   then bomb_y(i,j)=0
660   if j=1 and bomb_y(i,j)>256 then bomb_y(i,j)=0
670   if j=2 then bomb_x(i,j)=bomb_x(i,j)+8:if bomb_x(i,j)>=192 or bomb_y(i,j)>256
680   then bomb_y(i,j)=0
690   sp_set(42+i*3+j,bomb_x(i,j),bomb_y(i,j),&H122)
700 endfunc
```


X68000用
OMENS OF LOVEX1/turbo用
ENDLESS RAIN

X68000用 MUSICDRV サンプル曲 ©NAMCO

ダートフォックスより **Running up!**Kodama Kazuhiro
小玉 和博Fushiki Yoshihiro
伏喜 義宏Nishikawa Zenji
西川 善司

サンプリングは使用していません

X68000用に「OMENS OF LOVE」をお届けしましょう。この曲はフュージョンと呼ばれるジャンルの曲で、T-SQUAREが演奏しています。T-SQUAREは5人のグループで、カシオペアと並んで日本が世界に誇れるフュージョンバンドです。F1グランプリの曲、「TRUTH」などでおなじみですよ。

曲はインストなので、比較的FM音源だけのコンピュータでも作りやすい構成とは思いますが、テクニック命といっても過言ではないフュージョンを完全に再現するのは、かなり厳しいのではないのでしょうか。特にFM音源とは相性が最悪ともいえるようなギターが前面に出ている曲は至難の技だと思えます。

さて、作品のデキはといいますと、とっても気持ちいい曲になっています(?)。FM音源のみでAD PCMを使っていませんので普通のOPMDRV.Xのみで演奏できますが、サンプリングドラムに頼らなくても立派に演奏できるというお手本のような仕上がりです。もともとOPMAではボスコニアンサンプリングデータを使用していま

したので、あの元気なドラム達があるまま使われていました。そうすると静かな曲や、落ち着いた曲などではどうしてもドラムだけが浮いてしまっていたのです。そういう意味でも、この曲ではOPMのみで演奏したほうがきれいになるのです。試しにサンプリング対応にしてみたところ、やはりドラムが浮いていました。納得できない人は自分で試してみてください。

そういえば、フェードアウトもOPMだけならきれいにキマるといってもいいかもしれません。

XシリーズのX

X1用にはXの「ENDLESS RAIN」をお届けしましょう。Xはライブハウスからの叩き上げバンドです。自主制作していたアルバムが2枚あって、それがバカ売れたためレコード会社の目にとり、プロデビューに至ったという経歴を持っています。かなり正統派のバンドといえるでしょう。残念ながら矢板にあるSHARPさんのお抱えバンドではありません、悪しからず。

さて、作品についてですが、なかなか面白い構成をしているのではないのでしょうか。ヴォーカルをギターの音でやってしまったわりには、かなりまとまりがよいといえます。前述のとおり、ギターの音は結構ムズいのです。その分を考えるとよくできているといえます。

惜しむべきこととして、曲調を考えるとちょっとドラムの音が大きいのでは? と思えます。特にPSGのハイハットが怒鳴っています。確かに、原曲でははっきりと聞こえてはくるのですが……。ハイハットはノリを出すのにも使われますが、曲を引き

外は暑いようですが、皆さんいかがお過ごしでしょうか。さて、今月はT-SQUAREやXといったポピュラーソングものを2本と、MUSICDRVサンプル曲としてゲームミュージックを用意しました。ちょうど夏休みですし、打ち込んで聴いてみてください。また、100号記念としてMIDI基本テク特集も併設、ぜひ参考にしてください。



X

締めるのにも使われます。おそらく原曲の使い方は後者でしょう。引き締めるためのハイハットが、全体的に繊細な音で構成しているのを壊してしまうのはちょっともったいないですね。

やはりFM音源と比べてPSGの音質が落ちてしまうのは仕方ないことですので、PSGの使い方はしっかりと考えてみましょう。ソフトウェアエンベロープを掛けてコーラスラインとか、ハイハットならボリュームを小さめにするとか、S.E.を作ってみるなどが挙げられます。ミキサーをつないでいる人は、PSGの音量をFM音源の7割程度にしてみてください。あとは、好みに合わせてドラム系の音を気持ち下げて聴いてみてください。(S.K.)

「MUSICDRV」用サンプル曲

「MUSICDRV」用のサンプル曲のプログラムとして(注意:「OPMD」では演奏できません)、ナムコのトップビュータイプのオフロードカーレースゲーム「ダートフォックス」のメインテーマ「Running up!」をお届けします。この曲は、カシオペアの「Looking up」のパロディともいえる曲で(名前まで似ていたりする)、フュージョン



T-SQUARE

風のアレンジとなっていますからそっちの筋の方にもおすすめです。なんといってもウリはチョッパーの効いたベースと耳に突き刺すような高音のシンセソロ、右左にパンするバックিংです。作曲はもちろん(?)「メタルホーク」のめがてん細江氏です。

演奏方法

対応楽器はM1/R/Tシリーズ(以下M1)専用です。M1とMT-32の両方をお持ちの方はそのシステムに対応します(MT-32のみでは演奏できません)。また、FM音源も使用しているため、ミキサーなどでミキシングしてお楽しみください。まず、演奏させる前にM1側の設定をしてやります。

0) 「MUSICDRV」を、

A>MUSICDRV #180

のように組み込んでください。

1) 曲中で使用されている音色のベンドレンジを変更してください(2, 4, 46, 48, 51, 71, 72, 75, 92:後述の「MIDI基本テク特集」で説明してあります)。

2) M1リズムキット(音色番号09, 49)をEDIT PROG, F4-3「VDA1 KBD TRK」のパラメータを図Aのように設定します(リズムの音色をはかの音色よりも音量をやや大きくするため)。

3) GLOBALモードにしてDRUM KIT3の「TOM2」をすべて「TOM1」に、「OPEN

HH」と「CLOSED HH」のPANを(9:1)に変更してください(図B)。

4) 次にシーケンサモードにして、F1-4「MIDI CH」で各トラックのMIDIチャンネルを1, 11, 12, 13, 14, 15, 16, 10のように設定してください(図C)。

5) F4-1「TRACK PARAMETER」で各トラックのパンポットを(5:5), (5:5), (9:1), (1:9), (5:5), (8:2), (5:5), (5:5)のように設定してください(図D:プロテクトオフしてから設定すること)。

6) MT-32もお持ちの方は、M1のMIDI THRU 端子から MT-32の MIDI INへMIDIケーブルを接続し、MT-32の電源を入れてください。M1のみをお持ちの方は特に接続する必要はありません(当たり前だな)。

7) メインプログラムを入力、または入力されたものをロード、RUNしてください。
注意:MUSICDRVは7月号のデバッグ(バグを取ること)を行ったものを使用してください。さもないと、FM音源の音色が正常に鳴りません。

テクニックの解説

特に変わったことはしていませんが、ダンパーとピッチベンドを多用しています。ピッチベンドのMMLデータはbnd()という関数で作っています。たとえば、

bnd("c",12,8192,8875)

図A リズムキットの設定1

PROG I49 VDA1 KBD TRK Center Key
C-1 A+00 EGtime=0 AT:0 DT:0 ST:0 RT:0

図B リズムキットの設定2

DRUM KIT3 Closed HH1
#05 11 F#1 +009 L-57 D+00 9:1

図C MIDIチャンネルの設定

SONG1 MIDI CH
1G 11 12 13 14 15 16 10

図D パンポットの設定

SONG1 TRACK PARAMETRE
Tr 1 I01 V99 T+00 D+00 5:5 Prot: OFF

は、12個分の精度でピッチ8192(C)からピッチ8875(C+)まで滑らかに変化させるMMLを生成します。値、用語の意味については後述の「MIDI基本テク特集」を参照してください。

ところで、M1はコントロールチェンジにパンがありません。そのため基本的にはリアルタイムに音をパンすることが不可能です。しかし、各トラックにあらかじめ適当なパンを設定しておき、MIDIチャンネル切り替えコマンド「@n」で演奏チャンネルを切り替えることにより、パンをリアルタイムに切り替えているようなニュアンスを出すことができます。初心者の方M1ユーザーは参考にしてください。(善)

Oh!X通巻100号記念 MIDI基本テク特集

私が、サングラスをかけるとほとんどチンピラの西川善司です。6月号の創刊8周年記念のディスクに付いてきた「OPMD.X」と「MUSICDRV.X(サン・ミュージカル・サービス)」、ともに好評だったようです。両ツールともに、MIDI楽器をFM音源感覚のMMLで演奏可能なため、MML派の人間にとってはまさにたなからボタもちでしょう。

しかし、MIDI楽器はMIDI楽器。細かな表現をするのに大変重宝していた「Yコマンド」が使えないのははじめとして、MIDI楽器はFM音源やPSGとは違った箇所が多くあります。そこで、この場を借りてMIDI楽器を使うにあたっての基本テクを、音楽特集でもないのにババーンと公開してしましましょう。

MIDI楽器でディチューンをやる

FM音源(OPM)では、Y48+チャンネル番号(0~7)でピッチ(音程)を微妙にずらした音を重ねてやることによって、コーラス効果を実現できました。MIDI楽器には、こういったピッチをずらすコマンドはないのでしょうか。「ピッチ」という言

葉でピンときた読者もおられるでしょう。そうです、「ピッチベンド」のコマンドを用いるのです。

「MUSICDRV」では@B8192がピッチの基準値です。1チャンネルはこの基準値で鳴らしてやり、もう1チャンネルは基準値@B8192±50~100程度で鳴らしてやります(FM音源部もこの方法でコーラス効果を実現してやることができます)。

「OPMD」では基準値は128ですので1チャンネルは「Y9, 128」で鳴らし、もう1チャンネルは「Y9, 128±1~3」程度で鳴らしてやりましょう。

ここで注意がひとつ。楽器によってベンドの範囲が違うという点です。ROLAND MT-32は初期状態で1オクターブ範囲のピッチベンドが可能です。KORG M1/R/Tシリーズ(以下M1)では初期状態では半音範囲です。でもご安心を。たいいの機種は、このピッチベンドの範囲についてはコンフィギュレーションが可能です。M1の場合は音色単位でこの設定が可能です。音色を呼び出したあと、「EDIT PROG」モードにし、F7-2「JOYSTICK」(図1)の「P+02」を「P+12」にすることにより、MT-32のような1オクターブ範囲のピッチベンドが可能となります。そうそう、パラメータを

書き換えたあとはF9-1「WRITE/RENAME」(図2)で音色を再登録しなければいけません。まあ、ピッチベンドは1オクターブ範囲にしておいたほうが音色の応用範囲が広がるので、M1ユーザーはすべての音色をいまいった方法で変更しておきましょう。

ちなみに、ベンドの範囲を1オクターブにしたとき、「MUSICDRV」では半音が±683、「OPMD」では±11となります。つまり、「MUSICDRV」で、@B8192C@B8875C (8192+683=8875)

図1 JOY STICK

PROG I00 JOY STICK
P+00 F+00 PM00 MF0 FM00 MF0

図2 音色登録

PROG I00 A.PIANO Write/Rename
[<][>] [WRITE] → I00

とすると、最初の「C」はもちろん「C」ですが、2回目の「C」はC+で発音されます。「OPMD」で同じことをするには以下ようになります。

Y9,I28CY9,I39C (I28+I1=I39)

上の値をもっと細かいステップで与えて、各音を「&」でつないでやることにより「ポルタメント」を表現できます。

＆のお話

読者のハガキのなかにこんなのがありました。「OPMDでC&C+とする、とCの発音後C+の音が同時に鳴ってしまいます」。

FM音源では上のようにすると、Cの発音後、C+のアタック音なしに音程がC+へと変化します。「OPMD」では「&」はキーオフしないという目印に過ぎません。ですから、次にきたC+はCをキーオフせずに鳴ってしまいます。FM音源では1チャンネル1声という大原則があるので問題はないのですが、MIDI音源は1チャンネルで和音も発音可能なので、ハガキにあるような現象が起こるので

す。手抜きというより「OPMD」の性質上仕方のない現象なのです。しかし、後述の「ダンパー」効果を、この現象を逆手に取って実現できます。

ところで「MUSICDRV」では、ハガキにあるような例を演奏させると、

CC+

のように「&」が削除されたようなMMLが演奏されます。つまり以前に鳴ったキーコードとは違う音が新たに発音される場合、以前に鳴っていた音は強制的にキーオフされるわけです。また、FM音源部においても同様の処理が行われるので注意が必要です。

では、「MUSICDRV」や「OPMD」で「タイ」や「スラー」を実現するにはどうしたらよいのでしょうか。答えは簡単。先ほど、説明した「ピッチベンド」のコマンドを使ってやればよいのです。

@B8I92C&@B8875C MUSICDRV

Y9,I28C&Y9,I39C OPMD

おわかりいただけたでしょうか？

ダンパーってなんだー

「MUSICDRV」では「@d」というコマンドがあります。これは、「ダンパー」という機能を「オン/オフ」するもののなのですが、いままではFM音源のMMLのみを使っていた人にとっては耳新しい言葉です。言葉で説明するより例を用いて説明したほうがわかりやすいので、実際に「ダンパーコマンド」を用いて楽器を演奏させてみることにしましょう。

L16R@d127CEGR2.@d0

をいま演奏させたとしましょう。「L16」はただのデフォルト音長設定、続く「R」は16分休符となります。次の「@d127」はダンパーオンのコマンドで、以後発音される音はダンパーオンの効果がかかります。最初のCが発音され続いてEが発音されますが、このときCの音はキーオフされません。同様に最後のGもCとEが鳴った状態で発音されます(つまり、この時点ではCEGの和音が鳴っている)。さて、次に付点2分休符である「R2.」がきています。通常だと無音状態となるのですが、ダンパーオンの影響で「R2.」の時間、「CEG」の和音が鳴り続けます。そして、やっと最後の「@d0」

でダンパーオフとなり、発音されていた音はすべてキーオフされます。この例は譜面にするとちょうど図3のようになります。

「OPMD」でこれをやるには「&」を用いてやります。説明は「&」のところでしたので省きますが、上の例は「OPMD」では、

L16RC&E&G&G2.

となります。ただ「G2.」のあとにオールノートオフのメッセージを送らないと、CとEの音が鳴りっぱなしとなるので注意。

ベロシティのお話

これまた、FM音源から入ってきた人には耳新しい言葉です。MIDIの専門書などには「音の立ち上がり方の速さ」と書いてありますが、「ポリューム」のことだと思ってくださって結構です。いい方を変えれば「どのくらいの強さで鍵盤を叩いたか」ということです。ですから、実際の音量はVコマンドの値×このベロシティの値で決定されます。「MUSICDRV」で最大の音量で演奏するには、

@v127 @u127

を最初に送ってやります。

また、リズムマシンやポータブルキーボードのなかにVコマンドを認識しない機種があります(YAMAHA RX-8など)。そういった機種に対してはこのベロシティのみが音量を決定します。

@Lのお話

「MIDIドライバで@Lなどの微小音長を多用すると遅くなります」といった内容のハガキが届きました。うーん、FM音源を酷使する人は@LでガリガリとMMLを書く人が多いようです。そういえば、常連の立川正之君などは8分音符以上の音長は減多に書きません、なんていってました……。MIDIは31250bpsという、速いようで実はそんなに速くないボーレートで通信をしています。私の貧弱なMIDIシステムでも発音遅れはよくあることです。@Lの多用でテンポが遅れるなんてことは当然といえます。こういった問題の解決策としては、

- 1) @Lの使用を少し控える。
- 2) 内蔵FM音源に対してのみ@Lを使用する(内蔵FM音源は、MIDIで通信をしているのでなくI/OでMPUと直結しているの、かなり高速な応答が可能です)。
- 3) 適当なトラックのMMLの最初に「@LIR」を挿入し、割り込み周期のずれを作ってやる。

が挙げられます。

「MUSICDRV」のバグ

「MUSICDRV」にはバグがいくつか発見されています。「MVSET」コマンドで設定した音色番号とMMLの「@」コマンドの番号と対応しない、というバグは7月号で訂正されていますが、ほかのバグは取る手立てがないため(制作はサン・ミュージカル・サービスで、ソースリストはOh!X編集部にはありません)、これから話す解決方法で対処してください。

和音のコマンドは、

'CEG'

のように「」内の音を同時に鳴らすもので、最初の音に音長を書けばその長さで和音が鳴り続ける

図3 ダンパーの譜面



という、いままでのMMLの常識を破った大変便利なコマンドです。しかし、

'C8.EG'

のように付点を含む音長を記述すると、暴走してしまいます。これは、デフォルト音長設定コマンド「@L」を使ってやれば簡単に同様のことができます。つまり、付点8分音符なら、

@L36 'CEG'

です。では、全音符を超えた音長で和音を鳴らし続けるにはどうしたらよいのでしょうか(「@L」では全音符である192以上は記述できません)。

答えは「ダンパー」を用いて以下のようにしてやります。

@d127 'CIEG'RIRI@d0

この例だと、全音符3個分の長さで和音CEGが鳴り続けます(原理はすでにダンパーのところで説明したのでここでは省略します)。

「@n」はMMLトラックをMIDIチャンネルいくつに割り当てるかを演奏の途中で切り替える大変便利なコマンドです。「OPMD」では「Y4.?',「Y5.?'にあたります。「MUSICDRV」で以下のようなMMLを書いた場合、正常に動作しないので注意が必要です。

@n1@8.....@n2@8.....

順を追って説明すると、まず、@n1で現在演奏中のトラックをMIDIチャンネル1に変更し、次に音色切り替え「@8」で音色が切り替わります。「.....」は、まあ、MML演奏データがずらーっと並んでいるとして、これらはすべてMIDIチャンネル1で音色番号8で演奏されます。

さて、次に「@n2」がきているのでトラックをMIDIチャンネル2に変更します。問題は次の「@8」で、なんとMIDIチャンネル2へ音色切り替えのメッセージが送られないのです。どうも同じ音色は切り替えないというようなアルゴリズムのもとで動作しているらしく、しかもそれをMIDIチャンネル単位でなくトラック単位で行っているため、このような現象が起こるのでしょう。この対処方法としては、音色切り替え専用のトラックを設けるとか、別のトラックに音色切り替えのコマンドを挿入する、などが考えられます(ちょっと空しいね)。

いずれにせよ、サン・ミュージカル・サービスさんの迅速な対応が望まれますね。

MUSICDRVに望むこと

「MUSICDRV」はとてもよくできています。欲をいうと、ピッチベンドはオートベンドにしてほしい、ベンド、ベロシティやダンパー、モジュレーションなどの頻繁に使うコマンドは、できたら「@」という文字の必要のない1文字コマンドにしてほしい。あと和音のコマンドはFM音源にもほしかったなあ。サン・ミュージカル・サービスさん、Ver.2に期待しています、ゴロニヤーん。また、何かMUSICDRV楽器について質問があればどうぞ、Oh!X編集部・西川善司まで。

[illegible]


```

1920 b="eedr4ef&g":cs"eedr4ed&c2r>b<cd2.cde2.+tea4.ed&dr&k
1930 m_trk(3,+a+b+a+c":m_trk(5,+a+b+a+a)
1940 a="ccccccv4e4eeeeeeegggggggggaaaaaaac<d4ddddd+4c+c+c+c+c+c
+c+cccccccc
1950 m_trk(4,+a")bbbbggg<c&"+a")bbbbbbaa&a2..g+&g+2..g&g2..f+&
f+2..<f&")
1960 a="eedrrerd4dgrdr4.474v127dl+2@73v14r4.f4ferfr4f4farfr4.
474v127f1
1970 m_trk(6,+a"2b4.473v14e&"+a"2b4.473v14r4.a<ea&a2rcf<c&c2
>rcg<c&c2rea<c2v14&c")
1980 a="cc>br<cr4>b4b<dr>br4.474v127b-1a2a4.<v1473d4dc>rdr4c4
cfr4r4.474v127c1
1990 m_trk(7,+a"2d2g4.473v14c&"+a")b2g+4.v14aa&a2..g+&g+2..g&g2.
.aa&a2..473v13&")
2000 a="gggrgrg4gbrgr4.474v127gle2e4.473v14a4aararra4a<c>rar4
.474v127a1
2010 m_trk(8,+a"b2cd4.473v14g&"+a"2+<e4.v14e&e2..f+&f+2..e&e
2..f&f2..a&")
2020 /*
2030 a="fferderf4ferd>gr<c&":m_trk(1,a):a"o2@76a8a8<@75e8r8e8>
@76a8r8a8a4<@75ee8r>@76a8ra8a8"
2040 m_trk(2,a):a"fferdcfr4ferderc&":m_trk(3,a):m_trk(5,a)
2050 a="fferderf4ferd>g<rc8&":m_trk(4,a)
2060 a="cc>gragr<c4c>graar v120470e&":m_trk(7,a)
2070 a="aerfrea4aerffr v120470c&":m_trk(7,a)
2080 a="aagra4a4graar v1203@70g&":m_trk(8,a)
2090 /*
2100 a="1:16cccc:v15ccccv14cccc v13ccccv12cccc v11ccccv10cccc
v9ccccv8cccc v7ccccv6cccc v5ccccv4cccc v3ccccv2cccc v1ccccv0cccc
":m_trk(1,a)
2110 a="z+z+z+z"o2@76a<@75ee16f16e16>@76a16a&"+z+z+z+z+m_trk(2,a)
2120 a="v15"+z+z"v13"+z+z"v11"+z+z"v9o2@76a<@75ee16f16e16>@76a16a&
"+v7"z+z"v5"+z+z"v3"+z+z"v1"+z:m_trk(2,a)
2130 m_trk(3,"c2.c&c2.&c8r&"):m_trk(5,"c2.c&c2.&c8r&")
2140 a="effffe4derf4ferd4c&rdre4.drg&rff2.ef4ferd4ferd4c&rdcr":m_tr
k(3,"o4v13@74")a:m_trk(6,"e4.drg&rff2..+a)
2150 a="effffe4derv1l1f4fe4crdv9re4.drg&r7f&f2.ev4f4fe4derv2f4fe
4crvld":m_trk(3,"e4.drg&rff2..+a):m_trk(6,"e4.drg&rff2..+a)
2160 a="cc&c&c4b<rc4c&c4a>arbr<c4.b<drdc&c2.c&c4c4b<rc4c&c4a>a
rbr":m_trk(5,"o4v12@74")a:m_trk(7,"c4.")br<dr&c2.&c2..+a)
2170 a="o4cc&c&c4b<cvll1rc4c4a>arv9br<c4.b<dr7vc&c2.cv5c4cc4>
b<rcv3c4c4v2>arbv1r":m_trk(5,"c4.")br<dr&c2.&c2..+a):m_trk(7,"c4.")b
r<dr&c2.&c2..+a)
2180 a="cccccccccccccccccccccccccccc(v13crce)v10cccccccccccccccc
cccccccccccccccc":m_trk(4,a)
2190 a="v10ccccccccv8ccccccccccccccccccccccccccccv4ccccv6<crce>v4ccv2ccccce
cv1ccccccccv0cccccccccccccccc":m_trk(4,a)
2200 a="x4.grbra&a2.ga4ag4ggra4ag4frgrg4.grbra&a2.ga4ag4ggra4ag
4frgr":m_trk(8,a)
2210 a="v12g4.grbrv10a&a2.v8ga4ag4ggv6ra4ag4frgv4rg4.grbrv2a&a2
.ga4av1g4ggra4ag4frgr":m_trk(8,a)
2220 /*
2230 /* PLAY
2240 /*
2250 m_play()

```

日本音楽著作権協会(出)許諾第9070779-00

```

210 MEMS(&HB220+VR,36)=HEXCHR$("FA 00 51 23 73 11 23 25 4D 00 1F
1E 1D 1F 0F 0E 12 8B 04 04 04 06 A5 56 56 A5 00 80 80 00 00 00
80 00 00 00")
220 'SOUND NUMBER 6 STRINGS 1
230 MEMS(&HB244+VR,36)=HEXCHR$("FC 00 01 02 00 01 1F 1B 39 00 01
01 01 01 00 00 80 00 00 00 00 06 06 06 08 00 00 00 00 CC
80 00 02 00")
240 'SOUND NUMBER 7 HiHat OPEN 1
250 MEMS(&HB268+VR,36)=HEXCHR$("FC 03 33 3E 73 7F 00 00 07 00 1F
9F 9F 0C 11 1C 1E 03 C6 02 C7 32 54 32 54 00 00 00 00 00 C8
80 17 03 00")
260 'SOUND NUMBER 8 HiHat OPEN 2
270 MEMS(&HB28C+VR,36)=HEXCHR$("FC 03 30 3E 71 7F 00 07 0E 00 1F
9F 9F 0C 12 1C 1E 84 C6 C5 C7 32 84 32 74 00 00 00 00 00 C8
80 17 03 00")
280 'SOUND NUMBER 9 STRINGS 2
290 MEMS(&HB2B0+VR,36)=HEXCHR$("C4 00 64 24 68 68 2F 16 25 0F 1F
14 1F 14 00 00 00 00 00 00 00 00 06 06 06 00 00 00 00 00 C8
80 00 03 00")
300 'SOUND NUMBER 10 BASS
310 MEMS(&HB2D4+VR,36)=HEXCHR$("C2 00 70 52 20 30 1D 34 1B 00 19
0B 19 12 06 08 09 87 01 01 01 01 F8 FA FA F7 00 00 00 00 00 96
80 00 02 00")
320 'SOUND NUMBER 11 TOM
330 MEMS(&HB2F8+VR,36)=HEXCHR$("FB 00 05 00 01 01 04 0F 1B 00 1F
1F 1F 1F 1F 17 16 00 40 4C 0D 03 95 F8 08 00 00 00 00 00 00
80 00 00 00")
340 RUN"ENDLESS RAIN2.mml"

```

```

90 PLAY":";
100 RETURN
110 '='=====
120 '          VOCAL PART
130 '='=====
140 A(0)="I1 O3V9 Q8K0S2,2,0,11H3=1L8
150 A(1)="I6 O4V7 Q8 G1&G1&G1 V10
160 A(2)="G2G2 G2G2 G2G2 G2G2 C1 C2<B2> G2G2 G2G4.
```

```

170 A(3)="5G>EEEE16E16E4<GG>DDDD16D16&D4R4 RCCC16C16C<C<B&B16
>C16& C4D4C4<B4>
180 C="EEEE16E16E4R16<G16G16G16>DDDD16D16&D4R4
190 A(4)=<+>R4<AA>AAAA16G16& G4RDE4D4
200 A(5)="C4RED.C16&C<B>C4RDE4D4 C4RED4C<B16>C16& C2R2 R2R4.
210 A(6)=<G16G16>EEEE16E16E4R<G16G16>DDDD16D16&D4R4 RCCC16C16
&C<B&B16>C16& C4D4C4<B4>
220 A(7)=<+>R4.<A16A16>AAA16AA16 G4RDE4D4
230 A(8)="C4RD16D16EDC<B16>C16& C4RDE4D4 C4RED4C<B16>C16& C4R4R2
240 A(9)="E2.B4 C2R4D16C<B16> C2R4D16C<B16> C4RDC4<B4
250 A(10)="E4E4R4B4 C2R4D16C<B16> C2R4D16C<B16> C2<B2 I1003V11K
2
260 A(11)="C4S3,3,0,29H3=1G4=0<C4.D16&C16 F16&G4.&G16&G&F<S4,1,0
,6=1B4=0 A2.&AG16E16 F>FL16CFR32&G32RDR<B8GL8=1D=0
270 A(12)="C4S3,3,0,29=1G4=0<C4.G16&A16 S4,1,0,6=1G2=0<C4.<B> G1
6&A16&A4D16E16F.E.D16C16 <G>L16&C&DGD&A
280 A(13)=">E4D4
290 A(14)="C4RED4C<B16>C16& C4RDE4D4 C4RED4C<B16>C16& C4R4R2
300 A(15)=<+>C2R4C<B><C<B-1>>C2RCD<B- D4R4D4D4
310 A(16)="E-1 I2 V9 =00E32&F16.G0<E-FE-18'Q8DS3,2,0,6H3=1C=0L1
6<B>CDE32&F32E-DC<B- B-A-8.>=1C8=0<B-8A-GFE-FGA-B-L8> S3,2,0,4
=1C=0<B>=>1E=0D=1G=0F=1C=0<B-
320 A(17)="S3,2,0,6=1D>D16&=0E-16F16E-16D.<G16>DE-16F16 E-16&D16
CRD16&C16<B-4>F32&G16.F16E-16 <A-4.=1G16&=0A-16G4L16FGA-B- >C<A-B
><CD<B>>CDE-CDE-1A32&=0B-32GA-B-L8
330 A(18)=<+>R4G4FGA-B- G4E-16<B>DFA- G16F16E-4.DE-FA- G4>B-Q0<A-
B-A-18'>(GA-G)8'>(FGF)8'>(E-FE-18'>(DE-D)8'>Q8
340 A(19)=<+>C4F16G.FGA-B-16&A-16 G4FG<L16B>DFA-GFE-F E-FGA-B-GA-B
-><C<A-B>>CD<B>>CDL8
350 A(20)=<+>C2.Q0S4,1,0,5C32=1D&D32=0C16Q8'< B2.&B.>+A(0)+>G16> E
4E.F16&FER<G16G16> DDD16D16&D4R4
360 A(21)=<+>RCCC16C16&C<B&B> C4D4C4<B4> EEEE16E16E4R16<G16G16G16
> DDD16D16&D4R4
370 A(22)=<+>R4R<A16A16>AAA G4RDE4D4 C4RD16E16&EDC<B16>C16&
380 A(23)=<+>C4RDE4D4 C4RDE4D4<B16>C16& C4R4R2 R1
390 C="E2.B4 C2R4D16C<B16>
400 A(24)=<+>+>C4R4D16C<B16> C4C4<B4R4
410 A(25)=<+>C4R4D16C<B16> C4C4<B4R4
420 A(26)=<+>C4R4D16C<B16> C4RDC4<B4> 'Fade Out
430 A(27)=<+>E4E4R4B4 C2R.DC<B16> C2R.DC<B16> C2<B4R4
440
450 '!
460 DATA 0,1,2,0,3,4,5,6,7,8,9,10,11,12,0,13,14,9,10,0,15,16,17,
18,19,20
470 DATA 21,22,23,9,10,0,24,25,26,27,-1
480 '=====
490 ' VOCAL PART ECHO
500 '=====
510 A(0)=<+>I1 O3V5 Q8K2S2,2,0,18H3=1L8
520 A(1)=<+>I6 O4V7 Q8 D1&D1&D1 V9
530 A(2)=<+>E2D2 C2D2 C2<B2 A2B2 F1 F2E2 >C2C2 C2C4. R
540 A(6)=<+>CCCC16C16&C4RR<G16G16>DDDD16D16&D4R4 EEE16E16&EDC<B16
E16& E4F4E4D4R
550 A(8)=<+>C4RD16D16EEF16G16& G4RGG+4G+4 A4RAG4GG16G16& G4R4R2R
560 A(10)=<+>E4E4R4B4 C2R4D16C<B16> C2R4D16C<B16> C2<B2 I1003V7 K
2
570 A(16)=<+>E-2.&E I1 V8 =00E32&F16.Q0<E-FE-18'Q8DS3,2,0,6H3=1C=
0L16<B>CDE32&F32E-DC<B- B-A-8.>=1C8=0<B-8A-GFE-FGA-B-L8 S3,2,0
,4=1G=0F=1C=0<B>=>1E=0D=1G=0F=1C=0<B-
580 A(17)="S3,2,0,6>G4RA-16&G16F.<B-16>FG16A-16 G16F16E-16F16&D1
6D4A-32&B-16.A-16G16 C4.<=1B16&=0C16<B-4L16A-B>>CD E-CDE-FDE-FG
E-FGB-GA-B-L8
590 A(18)=<+>R4E-4DE-FG E-4<B-GGB>DF E-16D16C4.<B>>CDF E-4>GQ0<FG
F)8'>(E-FE-18'>(DE-D)8'>(CDC)8'>(B<C<B)8'>Q8
600 A(19)=<+>A-4>D16&E-DE-FG16&F16 E-4DE<L16GB>DFE-DCD CDE-FGE-
FGA-FGA-B-GA-B-L8
610 A(20)=<+>G2.Q0S4,4,0,5A-32=1B-&B-32=0A-16Q8' G1'+A(0)+>C4C.D1
6&DCRR<G16G16> DDD16D16&D4R4
620 A(22)=<+>R4R<A16A16>AAA G4RDE4D4 C4RD16E16&EDF16G16&
630 A(23)=<+>G4RGG+4G+4 A4RAG4GG16G16& G4R4R2 R1 R
640 C="C2.G4 A2R4B16AG16 A2R4B16AG16
650 A(24)=<+>A4A4G4R4
660 A(25)=<+>A4.AG4R4
670 A(26)=<+>A4RBA4G4
680 A(27)=<+>C4C4R4G4 A2R.BAG16 A2R.BAG16 A2G4R4
690 A(28)=<+>E4E4R4B4 C2R4D16C<B16> C2R4D16C<B16> C2<B4.
700
710 '!
720 DATA 0,1,2,0,3,4,5,6,7,8,9,10,11,12,0,13,14,9,10,0,15,16,17,
18,19,20
730 DATA 21,22,23,9,28,24,25,26,27,-1
740 '=====
750 ' Pianol
760 '=====
770 A(0)=<+>I5 O4V11Q8L8K0 R1R1R1
780 A(1)=<+>E<G>E<G>D<G>D<G> E<A>E<A>D<G>D<G>
790 A(2)=<+>AC<C>F<A>AC <AC>C<F>D4<G32>D32E16F
800 A(3)=<+>F<G>F<G>E<G>E<G> F<G>F<G>E<G>E<G>
810 C="E<G>CD<C>EGC D<G>BDBDF
820 A(4)=<+>E<A>CE<C>EAC F<A>F<A>D<G>D<G>
830 A(5)=<+>E<A>CEACAC D<G>D<G>E<G>D<G>
840 A(6)=<+>F<A>F<A>G<B>G<B> GCGCG<B>G<B>
850 A(7)=<+>ACACG<B>G<B> F<G>F<G>E<G>E<G> F<G>F<G>E<G>E<G>
860 A(8)=<+>A(4)
870 A(9)=<+>E<G>CE<C>EGC D<G>BDBDF E<A>CEACAC D<G>D<G>E<G>D<G>
880 A(10)=<+>ACACG<B>G<B> F<G>F<G>G16G16G
890 C="E<G>E<G>D<G>D<G> E<A>E<A>E<A>E<A> F<A>F<A>GCGC
900 A(11)=<+>ACACD<G>D<G>
910 A(12)=<+>ADADF<B>F<B>
920 A(13)=<+>E<G>CL16E8.CEGCEG<CL8 D<DGB>R16D16<D>GF E<EAB16><C<BA
B><C<BL8A FC.>E.D.<B><C<B16
930 A(14)=<+>L16>C2<C<EG>CE<G>CEL8 D2R16B>R16CD L16C4&CEC<A><C<AEA
EC4<E<A><C<AE>4L8 D4&D16D16<G>E<G>D<G>
940 A(15)=<+>F4<C16F16A>F.GDE16& E<G>E<G>E<G>+>D<G>+>C4<C16F16A16F
16>F.GB. >C4<F<G>E<G>CG<
950 A(16)=A(11)

```

```

960 A(17)=<+>E<G>E<G>D<G>D<G> E<A>E.>C16C<CAC ACACGCGC ADADG<B>G<B
>
970 A(18)=STRINGS(2,"A-4A-4>C4E-4< B-4B-.)>D16D.F16F4<")
980 C="G<B>>G<B>>F<B>>F<B>>E<G>E<G>D<F>D<F> E<A>E<A>E<A>E<G>
E<G>
990 A(19)=<+>C<A>>F<A>>F<B>>F<B>>
1000 A(20)=<+>F<A>>F<A>>E<A>>F<B>>
1010 A(21)=STRINGS(3,"E<A>>E<A>>F<B>>F<B>> G<B>>G<B>>G<B>>G<B>>")
)
1020 A(22)=<+>C<E>>C<E>>D<F>D<F>
1030 A(23)=<+>R<G>CDG<CDG D<GB>DB2< E<G>CE<C>C<E D<GB>DBDDF
1040 A(24)=<+>E<G>CE<C>EAC F<A>F<A>D<G>D<G> E<G>CE<C>C<E G<B>DG>
D<GDA-
1050 A(25)=<+>ACEA<C>C<F>C<F> C<D>C<DBDD ACACBDD
1060 A(26)=<+>C<E>C<EBE8 ACACBDD G4G4G4G16C.> C4C4C.C.<C<
1070 C="E<G>E<G>D<G>D<G> E<A>E<A>E<A>E<A>
1080 A(27)=<+>F<A>F<A>GCGC ADADF<B>D<G>
1090 A(28)=<+>F<G>F<G>GCGC ADADF<B>F<B>
1100 A(29)=<+>E<G>E<G>D<G>D<G> E<A>EC16C16C4A4 ACACGCGC ADADG<B>
G<B>
1110 A(30)=<+>ACACGCGC ADADG<B>G<B>
1120
1130 '!
1140 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23
1150 DATA 24,25,26,27,28,29,30,30,30,-1
1160 '=====
1170 ' String & PIANO ECHO & etc
1180 '=====
1190 A(0)=<+>I6 O3V9 Q8K0L8 C1&C1&C1 V11C1&C1&C1 <F1G1> C1&C1
I5 O4V7 Q8L8K2 R16
1200 A(10)=A(10)+>16
1210 A(11)=<+>S2,2,0,10H3=1I900V8 G1 A1 A2>C2 F2FED4
1220 A(12)=<+>E1 E1 F2G2 A2B2=0
1230 A(13)=<+>I5O4V7 R16E<G>CL16E8.CEGCEG<L8 D.<DGB16R32B16.D>DD C
R16<EAB16C<B&B>C<BL8A FC>C.<B.GGG16
1240 A(14)=<+>L16G2&G16EG<CE<G>CL8 <B2>R32G&G32R32G&G32B L16A4&AR
16>EC4<C<AEAC>4<EC<A>C<AE>4L8 D4&D16D16<G>E<G>D<G16
1250 A(15)=<+>C4R16<C16F16A16C.D.<B>C16& CR16<G>E<G>E<G>+>D<G+16 >
C4R16<C16F16A16C.DG. G4R16F<G>E<G>CG16
1260 A(16)=<+>S2,2,0,10H3=1I901V8 E2D2 C2D2 F2G2 A2D2
1270 A(17)=<+>E2D2 C2E2 A2G2 A2B2
1280 A(18)=<+>C2&C<A-B>>C D2.<B-4> C2&C16E-16GG F2F4=0I2 O3V10CD
1290 A(19)=<+>E-4A16&B-B-16&A-16GA-Q0<GA-G>8'Q8 S2,2,0,10=1I902V8
C2<B-2 A-2B-2> C2D2
1300 A(20)=<+>E-2F2 G2D2 C2E-2 F2<B-2
1310 A(21)=<+>I2 O4V9 =0A-2B-2 I902V8 =1B-2B2> C2D2 E-2D2
1320 A(22)=<+>C2D2< B-2B2 A-2B-2
1330 A(23)=<+>I2 O5V7 =0R16C2.Q0S4,1,0,5R16C32=1D&D32=0C16Q8'< B2.
AB I5O4V7 R16E<G>CE<C>C<G>C<E D<GB>DBDDF
1340 A(26)=<+>C<E>C<EBE8 ACACBDD16 >C4C4C4&C16E. F4F4E.E.E
1350 A(27)=<+>S2,2,0,10H3=1I901V8 E2D2 C2E2 F2G2 A2G4F4
1360 A(28)=<+>E2D2 A1 A2G2 A2B2
1370 A(29)=<+>E2D2 A1 F2G2 A2B2<CD<B
1380 A(30)=<+>C2<B2 >C1 F2G2 A2B2
1390 A(31)=<+>E1 E1 F2G2 F2G4D4
1400 A(32)=<+>C2<B2 >C1 F2G2 A2B2
1410
1420 '!
1430 DATA 0,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,2
3
1440 DATA 24,25,26,27,28,29,30,31,32,-1
1450 '=====
1460 ' Pianol
1470 '=====
1480 A(0)=<+>I5 O4V11Q8L8K0 R1R1R1
1490 A(1)=<+>CRCR<BRBR> >CRCR<BRBR>
1500 A(2)=<+>FRAR<CRFR> <FRARF4R64B32R64R16R
1510 A(3)=<+>CRRCRCRC CRRCRCRC<
1520 C="CRRCRCRC <BRBR>GR<BG>
1530 A(4)=<+>CRRCRCRC CRRCRCRC<BRBR>
1540 A(5)=<+>CRRCRCRC CRRCRCRC<BRBR>
1550 A(6)=<+>CRRCRCRC CRRCRCRC<BRBR>
1560 A(7)=<+>FRFRRCRC CRRCRCRC CRRCRCRC
1570 C="CRRCRCRC <BRBR>GR<BG>
1580 A(8)=<+>CRRCRCRC CRRCRCRC<BRBR>
1590 A(9)=<+>CRRCRCRC CRRCRCRC<BRBR>
1600 A(10)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1610 C="CRRCRCRC CRRCRCRC CRRCRCRC
1620 A(11)=<+>FRFRRCRC<BR>
1630 A(12)=<+>FRFRRCRC
1640 A(13)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1650 A(14)=<+>E2&E16R16RRR G2>D.E.G E4L16E>C<AE<AECEC<A>4><C<AEAE
C4 <B&B>BL8RBRBR
1660 A(15)=<+>A4R4A.BGG16& GR>CR<BRBR F4R4A.B>D. F4CRRC4.<
1670 A(16)=A(11)
1680 A(17)=<+>CRCR<BRBR> CRRCRCRC CRRCRCRC CRRCRCRC
1690 A(18)=<+>STRINGS(2,"C4C4E-4A-4 D4D.F16F-B-16B-4")
1700 C="E-RE-RDRDR CRRCRCRC<B-RB-R> CRRCRCRC<B-RB-R>
1710 A(19)=<+>CRRCRCRC
1720 A(20)=<+>CRRCRCRC
1730 A(21)=<+>STRINGS(3,"CRRCRCRC E-RE-RDRDR")
1740 A(22)=<+>A-RA-RB-RB-R
1750 A(23)=<+>I7O5V16D2R4.V15(BBB)8 B2I5O5V11G2< CR2RCR <BRRR>GR<B
G>
1760 A(24)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1770 A(25)=<+>ERRRRCRC CRRCRCRC CRRCRCRC
1780 A(26)=<+>GRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1790 A(27)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1800 A(28)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1810 A(29)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1820 A(30)=<+>CRRCRCRC CRRCRCRC CRRCRCRC CRRCRCRC
1830
1840 '!
1850 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23

```



```

1860 DATA 24,25,26,27,28,29,30,30,30,-1
1870 '=====
1880 ' Piano3
1890 '=====
1900 A(0)="I5 03V12Q8L8K0 R1R1R1
1910 A(1)="C1 C2.<C>G
1920 A(2)="F4.CF2 <G2G4BG
1930 A(3)=">C4.<G>C4.<G>C4.<G>C4.<G
1940 C=">C2.<CC C2.<C<B
1950 A(4)=C+"A2.<AA F4.FG4.G
1960 A(5)=C+"A2F4.F G4.GG4.G
1970 A(6)=">F4.FG4.G >C4.<G>E4.<B
1980 A(7)=">F4.CG4.D C4.<G>C4.<G> C4.<G>C4.<G>
1990 C=">C2.<CC C2.<C<B
2000 A(8)=C+"A2.<AA F4.FG4.G>
2010 A(9)=C+"A2F4.F G4.GG4.G>
2020 A(10)=">F4.FG4.G> C4.DE4.<B> F4.CG4.D C4.<G>GC16G16&GC
2030 C=">C4.C<B4.B A4.AG4.G
2040 A(11)=C+">F4.FE4.E D4.D<G4G4>
2050 A(12)=C+" F4.FE4.E D4.DG4G4>
2060 A(13)="C1 C2.<GG+ A1 F2G2>
2070 A(14)="C1 C2.<C<B A1 G2>G4G4
2080 A(15)=">F2G2 >C2<E2 F2G2 >C1
2090 A(16)=A(11)
2100 A(17)=">C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.C16F16EE4C1
6C+16 D4.F16F+16G4+G4
2110 A(18)=STRINGS(2,STRINGS(4,"A->E-<")&STRINGS(4,"A->D-<"))&"+>"
2120 C="E-4.E-D4.D C4.C<B-4.B- A-4.A-G4.G
2130 A(19)=C+">F4.FB-4.B->
2140 A(20)=C+">F4.FB-4B-4>
2150 A(21)=STRINGS(3,"<A-4.A-B-4.B-> E-4.E-<G4.G>")
2160 A(22)="<A-4.A-B-4.B->
2170 A(23)="<G1& G1> C2.<CC C2.<C<B>
2180 A(24)="<A2.<AA F4.FG4.G >C2.<CC C2.<C<B>
2190 A(25)="<A4.AF4.F G4.GG4.G F4.FG4.G>
2200 A(26)="<C4.CE4.E <F4.FG4.G >C4G>C<CE16G>C. C4.<G>CC4.
2210 A(27)="<C4.C<B4.B A4.AG4.G >F4.FE4.E D4.D<G4G4>
2220 A(28)="<C4.C<B4.B A4.AG4.G >F4.FE4.E D4.DG4G4>
2230 A(29)="<C4.C<B4.B A4.AG4G4 F4.FE4.E D4.DG4G4>
2240 '
2250 '
2260 DATA 0,1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23
2270 DATA 24,25,26,27,28,29,28,28,-1
2280 '=====
2290 ' BASS
2300 '=====
2310 A(0)="I1003V11Q8L8P3K0 R1R1R1
2320 A(1)=">S4,3,0,6H3=1C4.=0<G16&A16G4.C16&D16 C2F16&G.C16<G16
>C16&D16
2330 A(2)=">C16&D16&C2.< C2.<CF16&G16

```

```

2340 A(3)=">F2&F<A>CG16&A16 G2<S3,3,0,29H5=1G2=0 >C1& C1
2350 A(4)=">R1R1R1R1
2360 A(5)=">S4,1,0,6R4.>=1C&C16&=0<G.<=3G4=0
2370 A(6)=">C1 C2.<CG+ A1 F4.F16F+16G4<GA16G16
2380 A(7)=">C1 C2.<C<B A2>F4.F G2G.>D16&D4
2390 A(8)="><F4.F+GG4B >C4.DE4.E F4.F16F+16G2 C4.S3,3,0,40=3G=0>
C2
2400 A(9)=">C2<BB4B A4.AGG4E F2EE4E16C16 D4.AG4<GB>
2410 A(10)=">C4.G16C16<BB4G16G+16 A4.>AGG4G16E16 F4.C16F16EE4E16C
16 D4.F16F+16GG16D16<G4>
2420 A(11)=">C4H3=1G4=0<C4.D16&C16 F16&G4.<G16&G&F<S4,1,0,6=1B4=0
A2.<AG16E16 F>FL16CFRF32&G32RDR<B8GL8=1D=0
2430 A(12)=">C4S3,3,0,29=1G4=0<C4.G16&A16 S4,1,0,6=1G2=0<C4.<B> G
16&A16&A4>D16E16F.E.D16C16 <G>L16C&DGG&A&B&G&F&D16L8<
2440 A(13)=">F4.F16F+16GG4<B16B16> C4.D16D+16E4<E4 F4.F16F+16G4.G
> C4.S3,3,0,29=1G=0>C2<
2450 A(14)=">C4.<BB4.G A4.>AGG4D16E16 F4.C16F16EE4E16C16 D4.AG4<G
A16B16>
2460 A(15)=">C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.C16F16EE4C1
6C+16 D4.F16F+16G4+G4>
2470 A(16)=">A-A-A-A-A-A-A-A-
2480 A(16)=">C+">A(16)+>"+>A(16)+>"+>A-A-A-A-A-A-A-A-4
2490 C=">E-4.B-16E-16DD4<B-16B16> C4.G16>C16<B-B-4B-16G16
2500 A(17)=C+">A-4.E-16A-16GG4G16E-16 F4.>C<B-4<B->F16<B-16>
2510 A(18)=C+">A-4.>E-16<A-16GG4E-16E16 F4.A-16A16B-4<B-16BB-16>
2520 A(19)=">A-4.A-16A16B-B-4>C16D16 E-4.<F16F+16GG>G<G A-4.A-16
A16B-B-4>C16D16 E-4.F16F+16G4>B16>C16D<S4,1,0,6=1G<=0
2530 A(20)=">A-4.A-16A16B-B-4>C16D16 E-4.F16F+16G4+G4 A-4&L16A->E-
A-AB-FB-8B-8FF+L8
2540 A(21)=">G2.<G16>S3,3,0,40=1D.=0 G1< R1R1 R1R1R1R1
2550 A(22)=">S4,3,0,6H4=1>A2F2 G4G.<=0D16G16D16G16>D16<GG <F4.F1
6F+16GG4A16B16>
2560 A(23)=">C4.D16D+16E4<E4> F4.F16F+16G4=2G<0<G> C4&C16S3,1,0,8
=2G.=0<C4&C16&G.< >C4C16<S4,1,0,1=1G.=0C2<
2570 A(24)=">C4.G16>C16<<BB4G16G+16 A4.>AGG4D16E16 F4.>C16F16<EE4
C16C+16 D4.F16F+16G4<GB16G16>
2580 A(25)=">L16C4.GC<B8B4GG+ A4.>EAGAG4DE F4.>C<FE8E4CC+ D8>CC+D
8<FF+G8<G8G8AB>
2590 C=">C4.GC<B8B4GG+ A4.>EAG8G4DE
2600 A(26)=C+">F4.CFE8E4EC D8CC+D8.>CEDR<B8G<AB>
2610 A(27)=C+">F4.>C<AE8E4CC+ D8>CC+D<ADAG8D8<GABG>
2620 A(28)=">C4.G>C<B8B4GG+ A4.>A8G8G4DE F4.>C<FE8E4. D8A>D8<DA8
GD8.<G8AB>
2630 A(29)=">C4.GC<B8B4GG+ A8A8>A8.GAG4DE F8F4>CF<EC<G4<ECC+>4
D4.FF+=2G4=0<G4
2640 '
2650 '
2660 DATA 0,1,2,3,4,4,4,5,6,7,8,9,10,11,12,13,14,15
2670 DATA 16,17,18,19,20,21,22,23,24,25,26,27,28,29,-1
2680 '
2690 CHAIN"ENDLESS RAIN3.mml"

```

リスト4 ENDLESS RAIN3.mml

```

10 ' ' ENDLESS RAIN ' Program No.3
20 'SAVE"ENDLESS RAIN3.mml"
30 RESTORE
40 GOTO 100
50 LABEL "!"
60 READ I:IF I=-1 THEN 80
70 PLAY A(I):GOTO 60
80 PLAY"";
90 RETURN
100 'TOM ホーホー
110 T1="I11V11L64O3D&C<B&AV1200L8
120 T2="I11L64O2G&FA&E&DO0L8
130 T3="I11V13L64O2C&B&AGV1200L8
140 T7="I11V11L64O3D&CV1200L8
150 '=====
160 ' BASS ECHO & Drums
170 '=====
180 C="I4B4I3F
190 A(0)="I1003V11Q8L8P3K3 R1R1R1
200 A(5)=">R4.I11V13O3D64D32.<C32&<B32R16">T1+T2+">R16"
210 A(6)=T2+T2+T3+T3
220 A(7)=">O0V12S4,1,0,60I4BBR4I3F4R4
230 A(8)="I4BB16B16I3F.F16I4B16I3F16">T1+">R16">T2+">R16">T2+T3+T3
240 A(9)=C+"I4BBI3F4
250 A(10)=C+"I4B16B16R16B32B32I3F16F16R16F16
260 A(11)=C+"I4B16B16R16B16I3F4
270 A(12)=C+">F16I4B16I3F16R16">T1+">R16">T2+T2+T2
280 A(13)=C+"I4B16B16R16B16I3F16I4B16R16B16
290 A(14)="I4BBI3F16F16R16I4B32B32I3F16F16R16">T1+">R16">T2+T3+T3
300 A(15)="I4BBI3F4I4B16B16R16B32B32I3F16F16R16F16
310 A(16)=">L16I4B8B8I3F8.F8I4B13F4B32B32I3FFRFL8
320 A(17)=">L16I4B8B8I3F4B8R8B32B32I3F16I4B13F8I4B16B16
330 A(18)="I4BB16B16I3F.F16I4B16I3F16I4B16B16I3F.F16
340 A(19)="I4B16I3F16I4B16B16I3F.F16(RFFFFF)4I11V13O3=3<DD<GGCC>
400=0V12
350 A(20)=C+"I4B16B16R16B16I3F16I4B16B16
360 A(21)=C+">F16I4B16I3F16">T1+">R16">T1+T1+T2+T2
370 A(22)="I4B4R2R8I8O4V16<GGG>8 G2V12R2 O0R1 R1
380 A(23)="R1R1R1R1 I7O5V3 <AAA>8V6 <AAAAAA>4V8 <AAA>8V11<AAA>8V
12<AAAA>8V14<AAAA>8V15G16G16
390 A(24)="I7O5V15DI400V12B16B16I3F.I4B16I3F16">T1+">R16">T2+">R16
">T3+T3+T3
400 A(25)=C+"I4B16B16R16B16I3F.F16
410 A(26)="I4B.B16I3F.F16I4B16I3F16R16">T7+T7+T1+T2+T3+T3
420 A(27)=C+"I4BBI16B16I3F16
430 A(28)="I4B.B16I3F4F16F16R16F16">T1+T2+T1+T1
440 A(29)=C+"I4BBI16B16I3F4
450 A(30)="I4BBI16B16I3F.F16I4B16I3F16I4B16I3F16F16R16F16

```

```

460 A(32)=">L8I4B&BBI3FFFI4BI3FFFI11O3=3D<G=0I3O0FI11O2=3GC=0I3O
0FFFFFI11O3=3DD<GG=000L8
470 A(33)=C+">F16I4B16I3F16I4B32B32I3F16F16R16F16
480 A(34)="I4B.B16I3F.F16I4BBI3F.F16
490 A(35)=">L32I4B16I3FFFI4BI3F16I11O2=3AAAA=0I4O0B16I11O2=3DDDD=0I
400B16I11O1=3AAAA=00L8I4O0B11O1=3AAAA=000L8
500 A(36)=">L16I4B8BBI3FFRFI4BI3F8I4B32B32I3FFRFL8
510 A(37)="I4BB16B16I3F.F16I4BBI3F.F16
520 A(38)=">L16I4BBRBI3FFRFI4BI3F.R4I11O3=3DD=0
530 '
540 '
550 DATA 0,1,2,3,4,4,4,5,6,7,7,7,7,7,8,9,9,9,9,11,9,9
,9,12
560 DATA 9,9,9,11,9,9,9,13,9,9,9,14,9,9,9,13,9,9,9,15,16,17,18,1
9
570 DATA 9,9,9,20,9,9,9,13,9,9,9,11,9,9,21,22,23,24,9,9,9,25,26
580 DATA 9,9,9,13,9,27,9,28,9,29,9,30,9,16,32,9,33,34,35,9,36,
37,38,-1
590 '=====
600 ' PSG Hi Hat
610 '=====
620 A(0)=">V16_30L8y7,49y6,8S4,1,30,0^1=3
630 A(1)="R1R1R1R1
640 A(2)=">S4,3,20,0y6,4^C4_S4,1,30,0y6,6AAAAAA
650 A(3)=">y6,6AAAAAA AAAA
660 A(4)=">y6,6AAAAAA16A16=0y7,48Cy7,49=3
670 A(5)=">y6,6AAAAAA
680 A(6)=">y6,6AAAAAA4
690 A(7)=">y6,6AAAAA4R4
700 A(8)=">R4S4,3,20,0y6,4^C4_S4,1,30,0R2
710 A(9)=">S4,3,20,0y6,4^C4C4_S4,1,30,0R2
720 A(10)=">S4,3,20,0y6,4^C8.C4&C16C4R16C8.C4_S4,1,30,0
730 A(11)=">S4,3,20,0y6,4^C4C4_S4,1,30,0y6,6A16A16S4,3,20,0y6,4^C
4_S4,1,30,0
740 A(12)=">y6,6AAAAA=0y7,48C16C32C32C16C16y7,49=3R8
750 A(13)=">y6,6AAAAA4,3,20,0y6,4^C4_S4,1,30,0y6,6A4
760 A(14)=">S4,3,20,0y6,4^C4_S4,1,30,0y6,6AAS4,3,20,0y6,4^C4_S4,1
,30,0R4
770 A(15)=">S4,3,20,0y6,4^C2_S4,1,30,0R2
780 '
790 '
800 DATA 1,1,1,1,1,0,2,3,4,5,3,13,2,3,5,2,3,5,2,3,5
810 DATA 2,3,4,2,3,5,2,3,6,2,3,5,2,3,5,2,3,5,2,7
820 DATA 2,3,5,2,3,5,2,3,5,2,5,14,15,1,1,8,2,3,5,6
830 DATA 2,3,5,2,3,7,2,3,5,2,3,5,2,5,2,10,2,5,11,12,-1
840 '
850 PLAY "" :MUSIC START

```

リスト5 Running up!

```

00 /* */
20 /* */
30 /* Running up */
40 /* */
50 /* (C) NAMCO / MEGATEN */
60 /* */
70 /* Arranged by Z.NISHIKAWA */
80 /* */
90 /* */
100 m_init()
110 dim char v(4,10)
120 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.1
130 v=(59, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
140 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
150 31, 8, 1, 8, 7, 20, 2, 1, 5, 3, 0,
160 31, 8, 8, 7, 5, 24, 1, 2, 1, 1, 0,
170 31, 3, 7, 8, 1, 21, 1, 1, 3, 0, 0,
180 31, 0, 0, 9, 0, 0, 2, 8, 5, 2, 0)
190 m_vset(1,v)
200 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.2
210 v=(56, 15, 2, 1,100,127, 0, 4, 0, 3, 0,
220 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
230 20, 10, 0, 5, 0, 44, 0, 0, 0, 0, 0,
240 20, 16, 0, 10, 15, 27, 0, 0, 0, 0, 0,
250 20, 10, 0, 6, 15, 9, 0, 1, 0, 0, 0,
260 31, 18, 0, 10, 15, 0, 1, 13, 0, 0, 0)
270 m_vset(2,v)
280 /* AF OM WF SY SP PMD AMD PMS AMS PAN S.E.3
290 v=(59, 15, 2, 1,200,127, 0, 7, 0, 3, 0,
300 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
310 31, 8, 1, 8, 7, 40, 2, 2, 5, 3, 0,
320 31, 8, 8, 7, 5, 36, 1, 2, 1, 1, 0,
330 31, 3, 7, 8, 1, 29, 1, 14, 3, 0, 0,
340 31, 0, 0, 9, 0, 2, 2, 8, 5, 2, 0)
350 m_vset(3,v)
360 /* AF OM WF SY SP PMD AMD PMS AMS PAN GLOCKEN
370 v=(52, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
380 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
390 26, 3, 0, 2, 15, 35, 1, 6, 3, 0, 0,
400 31, 6, 0, 6, 15, 0, 1, 2, 4, 0, 0,
410 31, 6, 0, 1, 14, 41, 1, 10, 7, 0, 0,
420 31, 7, 0, 6, 15, 0, 1, 2, 7, 0, 0)
430 m_vset(4,v)
440 /* AF OM WF SY SP PMD AMD PMS AMS PAN FM SYNTH
450 v=(44, 15, 2, 1,200,127, 0, 0, 0, 3, 0,
460 /* AR DR SR RR SL OL KS ML DT1 DT2 AME
470 31, 0, 0, 0, 0, 22, 0, 2, 3, 0, 0,
480 27, 9, 0, 5, 1, 0, 0, 8, 3, 0, 0,
490 31, 0, 0, 0, 0, 23, 0, 4, 7, 0, 0,
500 27, 9, 0, 6, 1, 0, 0, 4, 7, 0, 0)
510 m_vset(5,v)
520 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256]
530 str j[256],k[256],l[256],m[256],n[256],o[256],p[256],q[256]
540 str s[256],t[256],u[256],v[256],x[256],y[256],z[256]
550 str a1[256],b1[256],c1[256],d1[256],e1[256],f1[256],g1[256]
560 str a2[256],b2[256],c2[256],d2[256],e2[256],f2[256],g2[256]
570 str j1[256],k1[256],l1[256],m1[256],n1[256],o1[256],q1[256]
580 str j2[256],k2[256],l2[256],m2[256],n2[256],o2[256],q2[256]
590 str s1[256],u1[256],v1[256],w1[256],x1[256],y1[256],z1[256]
600 str s2[256],u2[256],v2[256],w2[256],x2[256],y2[256],z2[256]
610 str u3[256],u4[256],a0[256],b3[256],h3[256],in[256],se1[256]
620 dim str q(4)[256],R(4)[256],S(4)[256],T(4)[256],U(4)[256],
V(4)[256]
630 sel=bnd("e",12,8192,16383)
640 for i=1 to 14:m_malloc(i,6000):m_assign(i,i:m_trk(i,"e99")
:next:m_alloc(11,12000):m_assign(11,11)
650 /*
660 key 2,"m_play()@M":key 12,"M_STOP()@M":key 19,"SAVE"+chr$(
34)+"RUN"+chr$(5):key 7,"m_trk("
670 m_tempo(182)
680 /* *** N O T I C E ***
690 /*EDIT PROG.
700 /* プログラム中で使われている全ての音色の
710 /* JOY STICK とのバレーパラメータを
720 /* JOYSTICK P+02 -> P+12
730 /* の様に変更して下さい。
740 /*SEQUENCER
750 /* MIDI CH 1,11,12,13,14,15,16,10
760 /* PAN Tr3=9:1 Tr4=1:9 Tr6=8:2 (Other Tr=5:5)
770 /*GLOBAL (DRUM KIT)
780 /* DRUM KIT 3の"Tom 2"を全て"Tom 1"にする。
790 /* OPEN HH and CLOSED HHのPANを共に9:1にする。
800 /*
810 /* TRACK 1 (BASS)
820 /*
830 a0="@n1 @47 o2 q8 @m0 @b8192 @d0"
840 in="@v12L8r4 @u120b-4f4(e-drc)r b-r4.a-4.r4agg-f4.& f1
850 as="@v12@u120L8l:b<b-a-gfgre-4<(e-d-c)>b<(cr)>b-4<b-a-gfgre-4
<(e-d-c)>d<@u127d->1|@u120e<@u127e->:|@u120e-16<@u127e->
860 x=bnd("e-",12,8192,0)
870 bl="@u120L8l:b<b-a-gfgre-4<(e-d-c)>b<(cr)>b-4<b-a-gfgl1lr4<e-
d-16>b-4<(e-a&L2)+x+@b8192L8>b-k:|re-4<(e-d-c-e-r4)>
880 bl="b<b-a-gfgre-4<(e-d-c)>d<@u127d->@u120e<@u127e-> t165
v14 r4a-2. t174(gge-e-)|t182
890 cc="@o2@u120L8l:b<b-a-gfgre-4<(e-d-c)>b<(cr)>b-4<b-a-gfgre-4|
1<(e-d-c)>d<@u127d->@u120e<@u127e->:|@u120q4<@u120q8d<@u127q6
d>@u120q8d<@u127q6d>@u120q8c<@u127q6c>
900 ds="L8o2q8a-(a-a-g-fe-fg-4a-(a-g-fd->)fe- >a-4<(a-g-fe-f>
a-4<(a-g-g-f(q6f)>q8<(q6g-> q8a-(a-g-g-fe-fg-4a-g->)f(f<d>-fe-
>a-4<(a-g-fe-fg-4a-g->-a-g-r-f(q6f)>q8<->q6g>

```

```

910 e="L8o2q8gfcgfcrg>g<gcf>b-<b-c(c)> g<gcfb>g>g<gcfb-afc
>g<gcfrcfg>g<gcf>b-<b-c(c)> g<gcfcf>g<gcfb-afc
920 g="L8o2q8b-<b-a-ga-b-r>b-4<b-a-ga-b-(e)-a- >a-<a>g<g>a-(e
->)rf b-<b>a-<a>b<g<ge-(e-) >b<b-a-ga-b-r>b-4<b-a-ga-b-(e)-a- >a-
<a>g<g>e-(e-b)-b- e-<a->ggf<fe<-e-
930 j="L8o2q8a-a-<g-a> >a-(a-g>a-4a-<g-a-e-fd-e- >b<g6b>q8g<
q6>q8a<q6a>q8a<q6a> q8b-<b>g<g>a-<a>f<f d-<d-c(c)>b<b>a-<a>
<a> f<f>g<g>a-<a>b<b> daa<d4>dad g<d>g<g4>q8dr
940 ke="L8v13o2q8l:6r1:r2. .b-4<b-a>a<q7gf4a-q8b>
950 l="L8l:3 [rb-rb-]1 :| r4<b-16a-16ga-b-r4> |:3 [rb-rb-]1 :|
rb-a-ga-gfg
960 m="L8v13@u12o2q8b-<b-a-gfg2gfgq6e-4<q2e-d-c>q7b<cr>>q8b-4
<b-a-gfg2gfgq6e-4<q2e-d-c>q7d<d-16>d-16e-(e-16 >b-4<b-r-gf
>f<fe-de-f4>b-4<b-a-gr<d-e-4 >q8f<q6f>q8e<q6e>q8e<q6e>>q8d<q6d
>
970 n=">b<b>a-16<a-16ga-frg e-<e>d-<d>a-<a-rg>> b-16<b-16
b-a->b->b<b>-fg b<'-e-a-'>'fb-'@u90'f16b-'g16(c)'@u120'f4b->>b<b>-
b<b>- b-16b-16<a-ge-f e-(e->)e-16e-16(c)>a-<a>gg- f<b-fa-4
gfb<-<d-q6d>q8c<q6c> q8b-<q6b>q8a-<q6a>-
980 o="@n1 @49 o4 q8 v1 @u127 @m0 @b8192L8 @d0r1 b-&v2b-&v3b-&
v5b-&v6b-&v8b-&v10b-&v12b-& b-1& @m127 b-1& @m0b-1& b-2&b-&v11b-
&v5b-&v7b-& t16v5b-&v3b-&v2b-&v1b-r2 t176@47r1t182
990 x="r>r>r2. r1 b-4fr<e-drc4>g>gr<ferd
1000 m_trk(1,a0+in)
1010 m_trk(1,a)
1020 m_trk(1,b):m_trk(1,b1)
1030 m_trk(1,"v12"+c)
1040 m_trk(1,d)
1050 m_trk(1,e)
1060 m_trk(1,d)
1070 m_trk(1,g)
1080 m_trk(1,g)
1090 m_trk(1,j)
1100 m_trk(1,k)
1110 m_trk(1,l)
1120 m_trk(1,m)
1130 m_trk(1,n)
1140 m_trk(1,g)
1150 m_trk(1,g)
1160 m_trk(1,j)
1170 m_trk(1,o)
1180 m_trk(1,"v13"+c)
1190 m_trk(1,d)
1200 m_trk(1,e)
1210 m_trk(1,e)
1220 m_trk(1,g)
1230 m_trk(1,g)
1240 m_trk(1,"v12"+j)
1250 m_trk(1,x)
1260 /*
1270 /* TRACK 2 (BRASS)
1280 /*
1290 a0="@n12 @3 q8 v7 @m0 @b8192 @d0"
1300 in="@u120L8o5r4'c4fb->'cfb-r'gb-<d>'gb-<c>'r'gb-<c>' r
gb-<d>'r4.'g-1b-ce->'@L72@d127'b-a<-ce-f>' r1@d0
1310 a="@u120L8o5l:'clfb-' r4.'ce-a-'r'ce-a-'r'ce-a-' 'clfb-' r
1:|
1320 bl="@clfb-'r4'r'ce-a-'r'ce-a-'r'ce-a-' <r1@L144'cfb->' L4@n
1163v09@u120'egb-<d>'egb-<c>'e-gb-<c>'e-gb-<d>'
1330 ce="@n11 @3 q8 v08 @m0 @b8192 @u110o5L8r1 r1 r1 r2@L72'gb-
ce->' r1 r1 L8r2..@ed127@u99'e-a-'>'b<-e-'>'gb-<'@L72>a-<d-'e-
g'
1340 d="@d0@u99o4q7L8r1 r1 r2'r'a-<d-f>'r4 r'a-<d-g>'r4'a<-d-f'
r4. r1 r1 r2..q8g- d->a<-f4d>a-[a-b<-d-e-]4
1350 e="@d0@u99o6L8r1 r2.q4'c4dfg' r2.'>cfb->'q6'cfb-' q4r2.'e
-4g(c)' r1 r2r8'>cdfg<'c4dfg' r2..@ed127>q8g fc4g416c16l16g16c
1360 g="@76 q8 o5 v13 @u120 @d0 r1 r1 r2..L16a-f d1 r1 r1 r2
b-ge-c>b-ge-c&
1370 h="cl r1 r1 @22o5'a-4<ce-f'r2. r1 r1 r1 r1
1380 j="@n11@3q6o4v07@u105@d0]:L8'a-<ce-g>'r'a<-cdf>'r'a<-ce-g>
'r'a<-cdf>'r 'a<-ce-g>'a<-ce-a>'r'a<-cdf>'red127a-<c'df'@d0] |
1'a-4<ce-g>'q7'fb<-ce>'a<-ce-a>'fb<-ce>'r@L72'fb<-cd' L2'ce-a-b-'>'cdfb
-' :| >a1<cdg'@L144q8'dfa-b'@r4
1390 k="@n12 @3 q8 v07"+a
1400 m="|:5r1:|@n11o5q4@u75L8r2d-e-4. r1 r1
1410 n="|:8r1:|"
1420 o="@n12 @13 q8 v08 @u110 @d0L8o5'clfb-' r4.'ce-a-'r'ce-a-'r
'ce-a-' 'clfb-' r1'+bl
1430 re="@d0@u99o4q7L8r1 r1 r2'r'a-<d-f>'r4 r'a-<d-g>'r4'a<-d-f'
r4. r1 r1 r2..q8g- d->a<-f4d>a-[a-b<-d-e-]4
1440 w="@n11@3q6o4v07@u105@d0]:L8'a-<ce-g>'r'a<-cdf>'r'a<-ce-g>
'r'a<-cdf>'r 'a<-ce-g>'a<-ce-a>'r'a<-cdf>'red127a-<c'df'@d0] |
1'a-4<ce-g>'q7'fb<-ce>'a<-ce-a>'fb<-ce>'r@L72'fb<-cd' L2'ce-a-b-'>'cdfb
-' :| >a1<cdg'@L144'dfa-b'>|
1450 x="@d0@u99o4q7L8r1 r1 r2.q4'c4dfg' r2.'>cfb->'q6'cfb-' q4r2.'e
-4g(c)' r1 r2r8'>cdfg<'c4dfg' r2..@ed127>q8g fc4g416c16l16g16c
1460 m_trk(2,a0+in)
1470 m_trk(2,a)
1480 m_trk(2,a):m_trk(2,b1)
1490 m_trk(2,c)
1500 m_trk(2,d)
1510 m_trk(2,e)
1520 m_trk(2,d)
1530 m_trk(2,g)
1540 m_trk(2,h)
1550 m_trk(2,j)
1560 m_trk(2,k)
1570 m_trk(2,a)
1580 m_trk(2,m)
1590 m_trk(2,n)
1600 m_trk(2,h)
1610 m_trk(2,h)
1620 m_trk(2,j)
1630 m_trk(2,o)
1640 m_trk(2,c)
1650 m_trk(2,r)
1660 m_trk(2,e)
1670 m_trk(2,r)

```



```

1680 m_trk(2,g)
1690 m_trk(2,h)
1700 m_trk(2,w)
1710 m_trk(2,x)
1720 /*
1730 /* TRACK 3 (PIANO)
1740 /*
1750 a0="en13 @72 o5 q8 v09 @m0 @b8192 @d0"
1760 in="u99L8o5r4 'c4fb->r4'gb-<d>'gb-<c>'r'gb-<c>' r'gb-<d>
>r4.'b-<1ce>'@L72@d127'a-<ce-f>' r1@d0
1770 a="u115L8o5l:'c1fb->r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb->r
1780 b1="c1fb->r4'r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb->' L4@n
12@52v07q6'egb-<d>'egb-<c>'e-gb-<c>'e-gb-<d>'
1790 c="L8l:'c1fb->r4'r'ce-a-'r'ce-a-'r'ce-a-'r'ce-a-' <r4'
4. :l r'>b-<e-g'r4.'g4b-<e-'r
1800 c1="c1fb->r4'r'ce-a-'r'ce-a-'r'ce-a-'r'ce-a-' <r4' 'c
fb->r4'r'ce-a-'r4.@d127a- e>b-<d-4.g4.
1810 d="e@d0l:'>b-<e-a-'r'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-
b'r4. 'b-<e-a-'r'>b-<e-g'r'>a-<d-f'r4 r'>a-<d-g-'r4'>a-<d-f'r4
'
1820 d1=">a-<e-a-'r'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-b-
'r4. 'a-<e-a-'r'>a-<e-g'r4'>a-<d-f'r r'>a-<d-f'r r'>a-<d-f'r r'>a-4<d-g-'r'd-g-
1830 e="e@d0l:'>g<cf'r'>g<ce'r4'cfb->'r |l'r'c2fa'cdf'gr:| r'cdf
a'r4.'e-g<c>'r
1840 e1=">g<cf'r'>g<ce'r4'cfb->'r r'c2fa'cdf'gr '>g<cf'r'>g<
ce'r4.@d127g fc4g4g16c16f16g16c>
1850 g=">:e@d0L72q8'cdfb->'@d127'a-1b-<df'L8r|@d0'fa-b-<e>'
'fa-b-<d>'@d127'e-1fa-<c>'r1l:|@d0'fa-b-<e>'fa-b-<f>'r@L168'e
-gb-'L8r@L127'e-1gb-<c>'r
1860 h=">:e@d0L72q8'cdfb->'@d127'a-1b-<df'L8r|@d0'fa-b-<e>'
'a-b-<d>'@d127'e-1fa-<c>'r1l:|@d0'fa-b-<e>'fa-b-<f>'r@L168'e
-gb-'r8@L144'e-gb-<c>'L8'a-<ce-g>'r'a-<ce-g>'
1870 j=">:L8'a-<ce-g>'r'a-<cdf>'r'a-<cdf>'r'a-<cdf>'r'a-<ce-g>
'>'a-<ce-a->'r'a-<cdf>'@d127a-<cdf>'@d0 |l'a-4<ce-g>'q7'fb-<ce-
>'fb-<ce>'@L72'fb-<cd' L2'ce-a-b-'cdfb->' :|>'a|<cdg'difa-
b'
1880 k="q8 v09o5"+a
1890 m="en11 @05 o4 q6 v08 @u110L8r'cdfb->'r4'cdfb->r4@L72'ce-fa-
-'L8'ce-fa-'r'ce-g'ce-fa-'r4 r'a-<ce-g'rql'>a-<ce-q4'>a-<ce-g>
'q1'c16e-fa-'>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q6'c4e-fa-'r@d
127L32dfgb-r2
1900 m1="e@d0L8q6r'cdfb->'r4'cdfb->r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'>a-<ce-g'r4'>a-<ce-g'r'ce-fa-'ce-fa-' @u90q2'
>b-4<dfb-'q1'>'f16b-<df'>'f16<f'>'f4b-<df'>'b-<dfa-'q8'>b-4<dfgb-
>'@u110
1910 n="q6r'cdfb->'r4q4'c4e-fa-'q2'ce-fa-'ce-fa-' q4'>a-4<ce-g'
'c4e-fa-'q4'c4dfb->'r4 rql'd32e-gb-'r32'd32b-de-gb-'de-gb-
-'r'de-gb-'r'de-gb-' q6'c4e-fa-'r8q8@L127L32cdfb-r2
1920 n1="e@d0L8q6r'cdfb->'r4'cdfb->r4@L72'ce-fa-'L8'ce-fa-'r'ce-
-g'q6@L72'ce-fa-' L4q7'ce-fb-'ce-g'ce-fa-'ce-fa-' L8@L127'b>b-
<'f'f'ce>'e-<e'>'d2d' //r8
1930 o="q8 o5 @u115 @d0 L8'c1fb->'r4.'ce-a-'r'ce-a-'r'ce-a-' 'c
1fb->'r1 'abl
1940 r1=">a-<e-a-'r'>b-<e-g'r4'>a-<d-f'r r'>a-4<d-g-'r'd-g-b-
'r4. 'a-<e-a-'r'>a-<e-g'r4'>a-<d-f'r r'>a-<d-f'r r'>a-<d-f'r r'>a-4<d-g-'r'd-g-
1950 w=">:L8'a-<ce-g>'r'a-<cdf>'r'a-<cdf>'r'a-<cdf>'r'a-<ce-g>
'>'a-<ce-a->'r'a-<cdf>'@d127a-<cdf>'@d0 |l'a-4<ce-g>'q7'fb-<ce-
>'fb-<ce>'@L72'fb-<cd' L2'ce-a-b-'cdfb->' :|>'a|<cdg'L8l:4r
'dfa-b'>:|
1960 m_trk(3,a0+in)
1970 m_trk(3,a)
1980 m_trk(3,a):m_trk(3,b1)
1990 m_trk(3,"en12 o6 q7 v07 @m0 @b8192 @u115"+c):m_trk(3,c
1)
2000 m_trk(3,d):m_trk(3,d1)
2010 m_trk(3,e):m_trk(3,e1)
2020 m_trk(3,d):m_trk(3,d1)
2030 m_trk(3,g)
2040 m_trk(3,h)
2050 m_trk(3,j)
2060 m_trk(3,"en13 @72"+k)
2070 m_trk(3,a)
2080 m_trk(3,m):m_trk(3,m1)
2090 m_trk(3,n):m_trk(3,n1)
2100 m_trk(3,"en12 @52 o6 q7 v07 @b8192 @u115 r8"+g)
2110 m_trk(3,h)
2120 m_trk(3,j)
2130 m_trk(3,"v09"+o)
2140 m_trk(3,"en12 o6 q7 v7 @m0 @b8192 @u115"+c):m_trk(3,c1)
2150 m_trk(3,d):m_trk(3,r1)
2160 m_trk(3,e):m_trk(3,e1)
2170 m_trk(3,d):m_trk(3,r1)
2180 m_trk(3,g)
2190 m_trk(3,h)
2200 m_trk(3,w)
2210 m_trk(3,x)
2220 /* MT-32
2230 m_trk(6,"en2 @4 o5 q7 v06 @m0 @b8292 @u90 p1"+in)
2240 m_trk(6,"|:20r1:|")
2250 m_trk(6,"o5"+c):m_trk(6,c1)
2260 m_trk(6,d):m_trk(6,d1)
2270 m_trk(6,e):m_trk(6,e1)
2280 m_trk(6,d):m_trk(6,d1)
2290 m_trk(6,g)
2300 m_trk(6,h)
2310 m_trk(6,"p2"+j)
2320 m_trk(6,"v6 o5 @p10"+a)
2330 m_trk(6,a)
2340 m_trk(6,"|:16r1:|")
2350 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
2360 m_trk(6,h)
2370 m_trk(6,"p2"+j+"p1")
2380 m_trk(6,o)
2390 m_trk(6,"en2 o5 q7 v06 @b8292 @u90 p1"+c):m_trk(6,c1)
2400 m_trk(6,d):m_trk(6,d1)
2410 m_trk(6,e):m_trk(6,e1)
2420 m_trk(6,d):m_trk(6,d1)
2430 m_trk(6,"o5 q7 v06 @b8292 @u90 p1"+g)
2440 m_trk(6,h)
2450 m_trk(6,"p2"+w+"p1")

```

```

2460 m_trk(6,x)
2470 /*
2480 /* TRACK 4 (PIANO 2)
2490 /*
2500 a0="en14 @3 o5 q8 v07 @m0 @d0
2510 in="r4 |:4r1:|
2520 a="e8b8292@u110L8l:'c1fb->'r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb-
->'r1:|
2530 b1="c1fb->'r4'r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb->' L4@n
13@72v08q6'egb-<d>'egb-<c>'e-gb-<c>'e-gb-<d>'
2540 c="L8l:'c1fb->'r4'r'ce-a-'r'ce-a-'r'ce-a-'r'ce-a-' <r4'
r4 :l r4'>b-4<e-g'r4'>'g4b-<e-'
2550 c1="r'cfb->'r4'r'ce-a-'r'ce-a-'r'ce-a-'r'ce-a-' <r4' r'
cfb->'r4'r'ce-a-'r4@d127a- e>b-<d-4.g4.
2560 d="e@d0r'>b-<e-a-'r'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r'd-g-
-b'r4 r'>b-<e-a-'r'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r4'>a-<d-f'r
4
2570 d1=">a-<e-a-'r'>b-<e-g'r4'>a-<d-f' r4'>a-4<d-g-'r'd-g-b-
-r4 r'>a-<e-a-'r'>a-<e-g'r4@d127g- d>a-<f4d>a-<a-b-<d-e>-l4
2580 e="e@d0l:'>g<cf'r4'>g<ce'r'cfb->' r4'c4fa'r'cdf'gr:r'>g<cf'r4'
>g<ce'r4@d127g fc4g4g16c16f16g16c>
2600 g="e8b8292l:e@d0L72q8'cdfb->'@d127'a-1b-<df'L8r|@d0'fa-b-<
e>'fa-b-<d>'@d127'e-1fa-<c>'r1l:|@d0'fa-b-<e>'fa-b-<f>'r@L
168'e-gb-'L8r@L127'e-1gb-<c>'r
2610 h=">:e@d0L72q8'cdfb->'@d127'a-1b-<df'L8r|@d0'fa-b-<e>'
'a-b-<d>'@d127'e-1fa-<c>'r@d0'a-1<ce-f>'>'fa-b-<e>'>'fa-b-<
f>'r@L168'e-gb-'r8@L144'e-gb-<c>'L8'a-<ce-g>'r'a-<ce-g>'
2620 j="e8b8192l8 r1:|
2630 k="q8 v07o5"+a
2640 m="en12 @05 o4 q6 v07 @u90 @b8292L8r32r'cdfb->'r4'cdfb->r4@
L72'ce-fa-'L8'ce-fa-'r'ce-g'ce-fa-'r4 r'a-<ce-g'rql'>a-<ce-q4'
>'a-<ce-g'q1'c16e-fa-'>a-16<ce-fa-'q4'>a-<ce-g'q1'ce-fa-' q6'c4
e-fa-'r@d127L32dfgb-r2
2650 m1="e@d0L8q6r'cdfb->'r4'cdfb->r4@L72'ce-fa-'L8'ce-fa-'r'ce-g
'@L72'ce-fa-' L8r'>a-<ce-g'r4'>a-<ce-g'r'ce-fa-'ce-fa-' @u70q2'
>b-4<dfb-'q1'>'f16b-<df'>'f16<f'>'f4b-<df'>'b-<dfa-'q8'>b-4<dfgb-
>'@u90
2660 o="en13 @3 q8 v07 o5@b8292@u110d0L8'c1fb->' r4.'ce-a-'r'ce-
-a-'r'ce-a-' 'c1fb->'r1'+abl
2670 r1=">a-<e-a-'r'>b-<e-g'r4'>a-<d-f'r r4'>a-4<d-g-'r'd-g-b-
-r4 r'>a-<e-a-'r'>a-<e-g'r4g- d>a-<f4d>a-<a-b-<d-e>-l4
2680 x="q8r1 r1 'c4fb->'L8'cfb->'r'gb-<d>'gb-<c>'r'@L72'dg<c>'
L8'dg<c>'>'ga<ce>'f+a<cd>'r'dfa<c>'
2690 m_trk(4,a0+in+a)
2700 m_trk(4,b1)
2710 m_trk(4,"en13 o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
2720 m_trk(4,d):m_trk(4,d1)
2730 m_trk(4,e):m_trk(4,e1)
2740 m_trk(4,d):m_trk(4,d1)
2750 m_trk(4,g)
2760 m_trk(4,h)
2770 m_trk(4,j)
2780 m_trk(4,"en14 @3"+k)
2790 m_trk(4,a)
2800 m_trk(4,m):m_trk(4,m1)
2810 m_trk(4,n):m_trk(4,n1)
2820 m_trk(4,"en13 @72 o5 q7 v08 @u115 r16"+g)
2830 m_trk(4,h)
2840 m_trk(4,j)
2850 m_trk(4,o)
2860 m_trk(4,"en13 @72 o5 q7 v08 @m0 @b8192 @u115"+c):m_trk(4,c
1)
2870 m_trk(4,d):m_trk(4,r1)
2880 m_trk(4,e):m_trk(4,e1)
2890 m_trk(4,d):m_trk(4,r1)
2900 m_trk(4,g)
2910 m_trk(4,h)
2920 m_trk(4,j)
2930 m_trk(4,j)
2940 /* MT-32
2950 m_trk(7,in+"|:20r1:|")
2960 m_trk(7,"en3 @1 o4 q7 v08 @m0 @b8292 @u99 p2"+c):m_trk(7,c
1)
2970 m_trk(7,d):m_trk(7,d1)
2980 m_trk(7,e):m_trk(7,e1)
2990 m_trk(7,d):m_trk(7,d1)
3000 m_trk(7,g)
3010 m_trk(7,h)
3020 m_trk(7,j)
3030 m_trk(7,"|:32r1:|")
3040 m_trk(7,"o4 q7 @b8292 @u99 p2"+g)
3050 m_trk(7,h)
3060 m_trk(7,j)
3070 m_trk(7,"|:8r1:|")
3080 m_trk(7,"en3 o4 q7 v08 @b8292 @u99 p2"+c):m_trk(7,c1)
3090 m_trk(7,d):m_trk(7,d1)
3100 m_trk(7,e):m_trk(7,e1)
3110 m_trk(7,d):m_trk(7,d1)
3120 m_trk(7,g)
3130 m_trk(7,h)
3140 m_trk(7,j)
3150 m_trk(7,x)
3160 /*
3170 /* TRACK 5 (BRASS 2)
3180 /*
3190 a0="en15 @73 q8 v11 @m0 @b8292 @d0"
3200 in="u120L8o6r4 'c4fb->'cfb->'r'gb-<d>'gb-<c>'r'gb-<c>' r
gb-<d>'r4.'g-1b-<ce>'@L72@d127'a-<ce-f>' r1@d0
3210 a="u110L8o6l:'c1fb->' r4.'ce-a-'r'ce-a-'r'ce-a-' 'c1fb->' r
1:|
3220 b1="c1fb->'r4'r'ce-a-'r'ce-a-'r'ce-a-' <r4@L144'cfb->' r1
3230 c="en15 @73 q8 v11 @m0 @b8292 @u110o6L8r1 r1 r1 r2r@L72'gb-
<e>'>'r1 r1 L8r2..@d127@u99'e-a-'>'>b-<e-'>'gb-<'@L72'>a-<d-'e-
-g'
3240 d="e@d0@u99o5q7L8r1 r1 r2r'a-<d-f>'r4 r'a-<d-g>'>'r4'a-<d-f'
r4. r1 r1 r2..@d127q8g- d>a-<f4d>a-<a-b-<d-e>-l4
3250 e="e@d0@u99o5q7L8r1 r1 r2q4'c4dfg' r2.'c1fb->'q6'cfb->' q4r2.'c
4e-g' r1 r2r8'cdfg'>'c4dfg' r2..@d127q8g fc4g4g16c16f16g16c>

```

```

3260 g="076 q8 o5 v13 @u120 @d0 @b8292r1 r1 r2..L16r32a-f d1 r1
r1 r1 r2b-ge-c>b-ge-c32&
3270 h="c11:7 r1 :|
3280 j="0n15073q6o5v11@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>
>'r'a-<cdf>'r'a-<ce-g>'a-<ce-a>'r'a-<cdf>'r@d127a-<c'df>@d0
|1'a-4<ce-g>'q7'fb-<ce->'fb-<ce->'r@L72'fb-<cd> L2'ce-a-b-'cdf
b-' :|>'a1<cdg>'d1fa-b'
3290 k="q8 v11"a
3300 m="1:8r1:|
3310 o="0n11049 o4 q8 v1 @u127 @m0 @b8292L8 @d0r1 b-&v2b-&v3b-&
v4b-&v6b-&v8b-&v10b-&v12b-& b-1& @m127 b-1& @m0b-1& b-2&b-&v11b-
&v9b-&v7b-& v5b-&v3b-&v2b-&v1b-v11r2 r1
3320 solo2()
3330 w="0n15073q6o5v11@u105@d0|:L8'a-<ce-g>'r'a-<cdf>'r'a-<ce-g>
>'r'a-<cdf>'r'a-<ce-g>'a-<ce-a>'r'a-<cdf>'r@d127a-<c'df>@d0
|1'a-4<ce-g>'q7'fb-<ce->'fb-<ce->'r@L72'fb-<cd> L2'ce-a-b-'cdf
b-' :|>'a1<cdg>L8|:4r'dfa-b'|
3340 x="q8r'dfa-b'r2. r1 'c4fb->L8'cfb-'r'gb-<d>'gb-<c>'r@L7
2'dg<c>'L8'dg<c>'r'ga<ce>'f+a<cd>'r'dfa<c>'
3350 m_trk(5,a0+in)
3360 m_trk(5,a)
3370 m_trk(5,a):m_trk(5,b1)
3380 m_trk(5,c)
3390 m_trk(5,d)
3400 m_trk(5,e)
3410 m_trk(5,d)
3420 m_trk(5,g)
3430 m_trk(5,h)
3440 m_trk(5,j)
3450 m_trk(5,k)
3460 m_trk(5,a)
3470 m_trk(5,m)
3480 m_trk(5,m)
3490 m_trk(5,g)
3500 m_trk(5,h)
3510 m_trk(5,j)
3520 m_trk(5,o)
3530 m_trk(5,c)
3540 for i=0 to 4:m_trk(5,R(i)):next
3550 m_trk(5,"073 q8 v11 @b8292@u110"+e)
3560 for i=0 to 4:m_trk(5,T(i)):next
3570 for i=0 to 4:m_trk(5,U(i)):next
3580 for i=0 to 4:m_trk(5,U(i)):next
3590 m_trk(5,w)
3600 m_trk(5,x)
3610 /*
3620 /* 00000 TRACK 8 (PERCUSSIONS)
3630 /*
3640 a0="0n16 @50 q8 @v127 @m0 @b8192 @d0"
3650 in="r4 @u110o6L4 ccc8c8r8c8 r8c8r.c8r r2r8c. r2.c
3660 a="@u110o6L4|:7crer:|crcc
3670 b="|:3crer:|r2c8r. |:4crer:|
3680 bl="crcc crcc rcr2 cccc
3690 c="@u110o6L4|:8crer:|
3700 g="L4|:8@u110o6c.@u120o3e8@u110o6c.@u120o3e8:|
3710 h="L4|:7@u110o6c.@u120o3e8@u110o6c.@u120o3e8:|@u110o6c.@u1
20o3e4@u110o6c8c
3720 j="@u110o6L4|:7crer:|cr.c8c
3730 k="@u110o6L4|:8crer:|
3740 l="|:3crer:|c.c8r2 |:3crer:| cr.c.
3750 m="@u110o6L4|:7crer:|crcc.c8c
3760 o="@u110o6L4|:6cccc:| r1 ccr.c8
3770 w="@u110o6L4|:7crer:|cccc
3780 x="cr2. r1 o3'd<ccc>'d'd8<ccc>'d'r8o6c8 rc(ccrc)2
3790 m_trk(8,a0+in)
3800 m_trk(8,a)
3810 m_trk(8,b):m_trk(8,b1)
3820 m_trk(8,c)
3830 m_trk(8,c)
3840 m_trk(8,c)
3850 m_trk(8,c)
3860 m_trk(8,g)
3870 m_trk(8,h)
3880 m_trk(8,j)
3890 m_trk(8,k)
3900 m_trk(8,l)
3910 m_trk(8,m)
3920 m_trk(8,m)
3930 m_trk(8,g)
3940 m_trk(8,h)
3950 m_trk(8,j)
3960 m_trk(8,o)
3970 m_trk(8,c)
3980 m_trk(8,c)
3990 m_trk(8,c)
4000 m_trk(8,c)
4010 m_trk(8,g)
4020 m_trk(8,h)
4030 m_trk(8,w)
4040 m_trk(8,x)
4050 /*
4060 a0="0n16 q8"
4070 in="0614@u99e16@u127e8. r1 r1 @u110o3r4c8>a8f8r4. r2.o6@L8
r@u99e@u110e @u115e@u120e@u127e
4080 a="@u127o6L4|:7rere:|re.e8.e8.
4090 b="|:3rere:| @u110o3c8>a16f8r8@u127o6e. |:3rere:|re.e8r
4100 bl="rere rer(reee)4 er.2. r1
4110 c="@u127o6L4|:7rere:| re.e8
4120 d="@u127o6L4|:8rere:|
4130 e="@u127o6L4|:7rere:| r8e8@u120eL8r@u120e@u127ee
4140 f="@u127o6L4|:7rere:| r8@u110o3c8c4>f8f8o6@u120e16e8e16
4150 g="@u127o6L4|:7rere:| rer8e8e8r8
4160 h="@u127o6L4|:7rere:| r8e8e8r8r8e8(eere)4
4170 j="@u127o6L4|:7rere:| |:@u110o3d@u127o6e:| |:rere:| @u110o
3d@u127o6ere @u110o3d@u127o6ee8@u110o2f15@u127o6e16r8e16e16
4180 k="@u127o6L4|:6rere:| rer8e8e r8e8er@u110o2f8r8
4190 l="@u127o6L4|:3rere:| L8ree16e16rer8r4 |:3r4e4r4e4:| @u110o
3roc>f4@u127o6e4
4200 m="@u127o6L4|:7rere:| re@u110o3r16c16c8>f8f8
4210 n="@u127o6L4|:7rere:| re@u110o3c16c16>a8a16f16f8
4220 o="@u120o3L8>far4f4.<c16c16 c>aa4.fr4 rar4f4.<c16c16 rec4>

```

```

r16a16af4 rar4f4.<c16c16 c>aa4.fr4 @u127o6e4e4r2 r2e4e4
4230 w="@u127o6L4|:3rere:| |:@u110o3d@u127o6e:| |:rere:| @u110o
3d@u127o6ere L8@u120o3d@u127o6e|:3re:|
4240 x="@u127r'e>>>d'r2. o3@u99c16c16@u110cc>aaffo6@u127e r4e4o
2f'f<<<e'ro3@u110d& d4d4ddr& d1
4250 m_trk(9,a0+in)
4260 m_trk(9,a)
4270 m_trk(9,b):m_trk(9,b1)
4280 m_trk(9,c)
4290 m_trk(9,d)
4300 m_trk(9,e)
4310 m_trk(9,f)
4320 m_trk(9,g)
4330 m_trk(9,h)
4340 m_trk(9,j)
4350 m_trk(9,k)
4360 m_trk(9,l)
4370 m_trk(9,m)
4380 m_trk(9,n)
4390 m_trk(9,g)
4400 m_trk(9,h)
4410 m_trk(9,j)
4420 m_trk(9,o)
4430 m_trk(9,c)
4440 m_trk(9,d)
4450 m_trk(9,e)
4460 m_trk(9,f)
4470 m_trk(9,g)
4480 m_trk(9,h)
4490 m_trk(9,w)
4500 m_trk(9,x)
4510 /*
4520 a0="0n16 q8"
4530 in="0u99o3L4r4 ddd8d8r8d8 r8d4.r8@u127d4. r2r8d4.& d1
4540 a="@u110o2L8|:3f+f+a+f+f+a+f+:|f+f+a+f+f+a+f+a+ 'f<d>'f
+a+f+f+a+f+f+:|f+f+a+f+f+a+f+f+:|f+f+a+f+f+a+f+f+a+@u110d.
4550 b="f<d>'f+a+f+f+a+f+f+:|f+f+a+f+f+a+f+f+:|r2f<@u120d4.>
@u110'f<d>'f+a+f+f+a+f+f+:|f+f+a+f+f+a+f+f+:| |f+r+f|
4560 bl="f<d>'f+a+f+f+a+f+f+: (f+a+f+a+)1 @u120<d1 L4|:@u110d
@u120d:|
4570 c="@u110o2L8'f<d>'f+a+f+f+a+f+f+:|3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+
4580 d="@u110o2L8'f<d>'f+a+f+f+a+f+f+:|3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+d>'a+f+a+
4590 e="@u110o2L8|:3f+f+a+f+f+a+f+f+: f+f+a+f+f+a+f+a+ :| f+f+a+
f+f+a+f+f+: L4f+a+f+a+8<d8>
4600 f="@u110o2L8'f<d>'f+a+f+f+a+f+f+:|3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+8<d8>
4610 g="@u110o2L8'f<d>'a+f+f+:|15f+a+f+f+:|
4620 j="@u110o2L8'f<d>'f+a+f+f+a+f+f+:|3f+f+a+f+f+a+f+a+ f+f+
a+f+f+a+f+f+:| L4f+a+f+a+
4630 k="@u110o3L8d1> r1 r1 r2f+f+16f+16f+a+ |:3r1:| r2f+f+16f+1
6f+a+
4640 l="<(rdrd)1 r4d4>f+f+16f+16'f<d>'a+ <(rdrd)1> r4<d4r4>f+f
+16f+16 <|:3(rdrd)1:| r1
4650 m="@u110o2L8|:56f+:|f+16f+16a+f+a+r2
4660 o="@u110o2L4|:24f+:| f+@u120'f<d>'r2 ddr2
4670 m_trk(10,a0+in)
4680 m_trk(10,a)
4690 m_trk(10,b):m_trk(10,b1)
4700 m_trk(10,c)
4710 m_trk(10,d)
4720 m_trk(10,e)
4730 m_trk(10,f)
4740 m_trk(10,g)
4750 m_trk(10,g)
4760 m_trk(10,j)
4770 m_trk(10,k)
4780 m_trk(10,l)
4790 m_trk(10,m)
4800 m_trk(10,m)
4810 m_trk(10,g)
4820 m_trk(10,g)
4830 m_trk(10,j)
4840 m_trk(10,o)
4850 m_trk(10,c)
4860 m_trk(10,d)
4870 m_trk(10,e)
4880 m_trk(10,f)
4890 m_trk(10,g)
4900 m_trk(10,g)
4910 m_trk(10,j)
4920 /*
4930 /* 00000 TRACK 11 (SE おまけです。)
4940 /*
4950 a0="0n17@1o3v12@u85q8@u0@b8192p3"
4960 a="@L1|:2 r2."+"sel+r8. r2."+"sel+r16":a1=sel+"r16 r2."+"se
1+r8."+"a2="r2r8"+"sel+r8."+"sel+r16:|
4970 b="|:2 r2."+"sel+r8. r2."+"sel+r16":b1=sel+"r16 r2."+"sel+r
8. r1:|
4980 b2="r2."+"sel+r8. r2."+"sel+r16":b3=sel+"r16r1r1"
4990 c="|:7L16@2o4q8v12@u115@b8192 p3<er>er<er>er r8@1v12@u95o3
p3
5000 x=bnd("e",12,8192,15705)
5010 k1="0L1"+x+r8."+x+r16"
5020 k2="02@v127@u127@b8192o4p3e8p2c8>a4o3o2v11@u80@L3p3e&f+f&
g&g+&aa&a&b&b&c&c&+&d&d&+&p&e&f+f&g&g @u70g+&aa&a&b&b&c&c&+&d&d&+&u60
e&e+&f&g&g+&aa&a&b&b :|
5030 m="1:8r1:|
5040 o="|:3L16@2o4q8v12@u115@b8192 p3<er>er<er>er r8@1v12@u95o3
p3
5050 m_trk(11,a0+"r4 |:4r1:|")
5060 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5070 m_trk(11,b):m_trk(11,b1):m_trk(11,b2):m_trk(11,b3)
5080 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5090 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5100 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5110 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5120 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5130 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)
5140 m_trk(11,a):m_trk(11,a1):m_trk(11,a2)

```


●リンクWLK

今月はリロケータブルファイルを実行可能なマシン語ファイルに変換するリンクWLKの登場です。マシン語プログラムを開発する場合には先月発表したWZDと今月のWLKの2つが必要ですので注意してください。WZDとWLKはいわば表裏一体で、どちらが欠けても用をなしません。

それならば、なぜまとめてひとつのプログラムにしてしまわないのかと疑問を持たれるかもしれませんね。リロケータブルファイルはマシン語ファイルの一種ですが、CALL命令やJP命令の飛び先などのアドレスがまだ確定していない不完全なマシン語ファイルです。

逆にリロケータブルファイルのこの特性を利用すると、CALL先がファイル内になくてもアセンブルできるというメリットが生まれます。これらの不完全なマシン語ファイルをいくつかつなぎ合わせ、未確定のアドレスなどを確定して完全なマシン語ファイルにするのがリンクの役割です。

リンクの役割はリロケータブルファイルを実行可能なマシン語ファイルにすることだけであり、そのリロケータブルファイルがどんな処理系によって出力されようと思ったことではないのです。

第97部

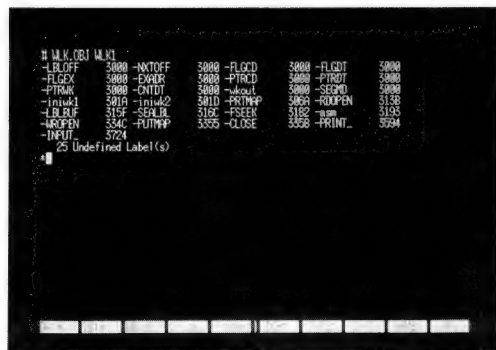
リンクWLK

リロケータブルファイルを出力するのはCコンパイラかもしれないし、PASCALコンパイラかもしれない。JP先やCALL先がプログラム内になくてもいいのですから、メインとなる部分はC言語を使って書き、速度を要求される部分や緻密な処理を行いたい部分だけをアセンブリ言語で書くという技も使えます。リロケータブルファイルを扱う世界ではシステムプログラムはリロケータブルファイルを出力するものとそれを実行可能なマシン語ファイルに交換するものの2つに大きく分けることができるといえるでしょう。

WZDは前者に当たります。数ある(今後登場するかもしれない)リロケータブルファイルを出力するもののひとつにすぎません。そしてWLKは後者です。表裏一体となって使われる2つのプログラムがひとつにまとめられていない理由がおわかりいただけたでしょうか。

●四方山話

編集室に面白いゲームが届きました。編集者とライターの間で対戦が盛り上がっています。一世を風靡したこのゲームを来月はお届けすることにし、ライブラリアンは再来月ということになりました。ご期待ください。



●S-OSの系譜 (12)

S-OS "SWORD" はCP/MやMS-DOSなどのDOSと比較して足りない機能がいくつか存在していました。ひとつは何度か書いたようにファイルの扱いが弱いということです。ファイルとの1文字単位の入出力はサポートされていませんし、ファイル内をシークするなんてこともできません。しかし、こういった機能をほしい人(つまりファイル処理を行うS-OSのアプリケーションを自分で作成しようとする人)なら、用意されている機能を使って好きなように実現できる機能ですから、声高に叫ぶほどのことではないでしょう。

それより、掲載されたプログラムを入力して使う人にとってはユーザーフレンドリでないという点のほうが問題だったのではないのでしょうか。CP/MやMS-DOSがユーザーフレンドリだと思う人はいないでしょうが、S-OSのユーザーインターフェイスはそれに輪をかけて冷たいものでした。

そもそもユーザーインターフェイスなどと呼べるシェルは存在せず、モニタという名のコマンド解釈プログラムが用意されているだけだったのだから当然です。プログラムを実行するには、まずマシン語ファイルをロードし、実行開始番地へジャンプする。まさにマシン語モニタ感覚です。

1986年10月号では、このモニタを機能拡張しようという試みが行われました。まず、マシン語ファイルのロード・実行を一気に行うRコマンドをモニタに追加。さらに、常用できる小さなプログラムとして、ファイルのコピープログラム(S-OSではASCIIファイルのコピーができない)とファイルの内容を画面に表示するプログラムが発売されました。これにより、モニタの使い勝手も随分向上しました。この試みは1987年5月号のS-OS "SWORD" 変身セットへと受け継がれていくことになります。

同時に発表されたディスクモニタDREAMは、ディスク上のファイルを連続クラスタに収め直す、ディレクトリを並べ替えるといった便利な機能を備えたツールとして人気を集めました。

リンカWLK

Ishigami Tatsuya

石上 達也

お待たせしました。ついにリンカ WLK の登場です。これで WZD でアセンブルしたオブジェクトファイルを実際に起動できるファイルにすることができます。これら WZD, WLK は S-OS に新しい開発環境を築いてくれることでしょう。

WLKとは?

皆さん先月号のWZDは、もう入力し終わったでしょうか? なにせ9Kバイトにも及ぶ超大作だったので大変だったでしょう。入力するのも大変ならデバッグするのは、その10倍大変なのです(システム関係は、ゲームと違いデバッグ=気分転換にならない)。できるだけ多くの人に使っていたきたいものです。

さて、その9Kバイトにも及ぶWZDとともに大規模アプリケーションの開発に威力を発揮するのが、このプログラムです。ネーミングはWZD専用のリンカということで、WLKとしました。とくにアルファベットの語呂合せは、考えていません。今回は約7Kバイトの分量です。がんばって入力してください。

リンカとは?

先月号でも少しお話ししましたが大規模なプログラム開発には、リロケータブルアセンブラを用いると有利です。そのとき、リロケータブルアセンブラから出力された個々のリロケータブルファイルをつなげ(リンクし)1本のオブジェクトファイルを出力するのがこのリンカです。

ざっと使い方を説明しましょう。プロンプト' * 'が出ている状態が、入力可能状態です。ここでは、WZDのリンク作業を例にとって説明します。

まずWZDのメモリ配置は、3000_H番地からコードセグメント、6000_H番地からがデータセグメントですので、それぞれのスタートアドレスをリンカに知らせてやりま

す。*/P:3000,/D:6000と入力します。そして、WZD本体は、WZD1,WZD2,WZD3,WZD35,WZD4から成っていますので、これらをリンクします。

*WZD1

ここまで入力し終わったなら、画面上には、なにやら表が表示されるはず。これは、WZD1をリンクした時点での、未定義ラベルとそのラベルが初めて使用された場所です。次に、

*WZD2,WZD3,WZD35,WZD4

と、入力してください。今度はなにも表示されずに、プロンプト' * 'が表示されたはず。なぜなら、WZDのすべてのファイルをリンクし終えたので、未定義のラベルはもうないからです。

この状態で、もしラベルリストがほしいれば、

*WZD/M

と入力してください。ファイル“WZD.MAP”にラベルリストが収録されています。

次に、WZDはデータセグメントに初期条件などを置いていないので(つまり、実際に必要なのはコードセグメントのみなので)、

*WZD/N:P

と入力すると、“WZD.OBJ”というファイル名で、オブジェクトファイルを作ります。

*WZD/N

と、最後にやってしまうとデータセグメントに割り振ったワークエリア領域までも含んだファイルができてしまいます。これでも一応は動くWZD.OBJができるのですが、かなりの無駄な部分まで含んでしまいます。

と、こんな具合にリンク作業は行いますが、要は、/P:スイッチと/D:スイッチで、スタートアドレスを決めてやり、リンクしたいファイルの名前を打ち込んでやればよいのです。そうしたら、/Nスイッチ

で、それらをオブジェクトファイルとして、取り出せます。

プログラム

このプログラムもWZDと同様に腕力にものをいわせて作ったものです。それぞれのアイテムに応じて、適当な処理を行います。サブルーチンも何カ所か同じようなものが出てきたら、適宜作っていくという感じ。WZDと違いリンカは、これでも作れた。

サブルーチンごとに、ほとんどが独立しているので解析はそんなに困難ではないと思います。

WZDと同様にWLK1.ASMは、最初Small-Cを用いて作成しました。このとき、

- 1) ローカル変数は、だいたい1関数につき3つ以内(BC, DE, HLレジスタに対応させることができる)
 - 2) グローバルなポインタは2つ以内(IX, IYレジスタに対応させることができる)
- という条件が満たされていたのでハンドコンパイルは、たいへんスムーズに行えました。

気になるサイズですが(プログラムのサイズです。念のため)、だいたいSmall-Cでコンパイルしたときの1/3程度になりました(ライブラリは除いて。本体のみ)。

メモリマップの変更は、リンカに与えるパラメータとWLKのヘッダファイルであるWLK.Hの内容を変更することによって行えます。現在、ローカルなラベルは1024個使用できるようになっていますが、この数を変更するときもこのWLK.Hを書き換えます。

WLKをソースからアセンブルするとき

```
# WZD
*=WLK1
*=WLK2
*=WLK3
* [ここでシフト+ブレイクを押す]
# WLK
*/P:3000,/D:4500
*WLK1,WLK2,WLK3,WLK/N:P
とすれば、ここに掲載されているものと同様なオブジェクトが得られます。
```

いうまでもないと思いますが、ソースリストのみを一生懸命に入力しても、WZDとWLKのオブジェクト形式のプログラムがなければ、アセンブラを通してオブジェクトを得るということはできません。まず最初に、WZDとWLKのオブジェクト・プログラムが絶対に必要です。

最後にばらしてしましますが、べき乗を行うサブルーチンは、第1パラメータ(Hレジスタ)を、第2パラメータ(DEレジスタ)回掛け合わせるということをしていきます。本当は、

$$A^B = \text{EXP}(B * \text{Log}(A))$$

を展開して、ゴリゴリ計算したかったのですが、メモリを大量に使ううえ、浮動小数のレベルで計算しなければ精度が出ないので、しかたなしに、中学1年生しています。腕に自信のある方、なにかよい方法をご存じの方はご連絡ください。

WZDの訂正

コマンドラインからのRUNコマンドを拡張した“SWORD”を使用した場合、処理を終了して“SWORD”のモニタに戻ってくるときに誤動作する場合があります。

```
3008 ED 5B 76 1F → CD AB 50 00
303A 2A 76 1F → CD B2 50
3153 ED 7B 00 → C3 FA 1F
347A ED 7B 00 → C3 FA 1F
```

以上のように訂正したうえ、次のダンプリストのようにプログラム(50B7Hまで)を追加してください。

```
5080 3E 09 37 18 04 3E 80 90 : E8
5088 A7 E1 C1 C9 26 00 6F 29 : D0
5090 29 29 29 C9 E5 CB 3C CB : FB
5098 1D CB 3C CB 1D 7D E1 C9 : 33
50A0 1D CB 3C CB 1D 7D E1 C9 : 33
50A8 00 00 00 ED 5B 76 1F 13 : F0
50B0 13 C9 2A 76 1F 23 23 C9 : AA
50B8 00 00 00 00 00 00 00 00 : 00
50C0 00 00 00 00 00 00 00 00 : 00
50C8 00 00 00 00 00 00 00 00 : 00
50D0 00 00 00 00 00 00 00 00 : 00
50D8 00 00 00 00 00 00 00 00 : 00
50E0 00 00 00 00 00 00 00 00 : 00
50E8 00 00 00 00 00 00 00 00 : 00
50F0 00 00 00 00 00 00 00 00 : 00
50F8 00 00 00 00 00 00 00 00 : 00
-----
SUM: 5B 72 C3 A3 C3 EA 8A F4 5C85
```

お詫びと訂正

実は、先月号の私の記事に少しばかり誤りがありましたので、ここに訂正させていただきます。まず、プログラムリストの呼び方が、記事中と注記中では、異なっていたこと。注記中のリスト2から6までは、それぞれ順に、

WZD1,WZD2,WZD3,WZD35,WZD4
です。

また、79ページの中段31行目を、

表1

■起動方法

“SWORD”の拡張をしていない人はコマンドラインから、
#LWLK
#J3000
と、拡張をしてある人は、
#WLK
で起動します。すると、この後ろにプロンプト‘*’を表示してパラメータの入力待ちになります。なお、それぞれ、
#J3000
#WLK
の後ろにパラメータを書くことができます。

■パラメータ

以下、[ファイル名]とあるのは、大文字小文字を区別し、スイッチ(/Sとか/Nとか)は、どちらでもかまいません。

* [ファイル名]

[ファイル名]で表されたリロケータブルファイルを取り込みます。省略時の拡張子は‘.REL’です。

* [ファイル名] /S

[ファイル名]で表されたライブラリファイルを、未定義なモジュールに限り取り込みます。省略

*WZD1,WZD2,WZD3,WZD35,WZD4,
WZD/N:P

としてください。

* * *

6月16日のコンサートにきてくださった皆様ありがとうございました。おかげさまで、コンサートは大成功をおさめることができました。ちなみに、コンサートのプログラムの役員紹介の写真で、コンサート委員長をひっくり返して下から支えているのが私です。来年もやりますので、ぜひ、またきてください。

時の拡張子は、‘.LIB’です。ライブラリファイルとかモジュールとかは、ライブラリアンWLKのときに詳しく説明します。

* /P:xxxx

コードセグメントの領域を、アドレスxxxx(16進数で4桁以内)から取ります。

* /D:xxxx

データセグメントの領域を、アドレスxxxx(16進数で4桁以内)から取ります。

* [ファイル名] /N

リンクの結果を[ファイル名]で表されるファイルにコードセグメント、データセグメントともに出力します。デフォルトの拡張子は、‘.OBJ’です。以下同様。

* [ファイル名] /N:P

リンクの結果を[ファイル名]で表されるファイルにコードセグメントのみ出力します。

* [ファイル名] /N:D

リンクの結果を[ファイル名]で表されるファイルにデータセグメントのみ出力します。

* [ファイル名] /M

ラベル情報を[ファイル名]で表されるファイルに出力する(外部ラベルの情報のみ)。

シフト+ブレイク

リンカの作業を中断し“SWORD”のコマンドラインに戻ります。

表2 エラーメッセージ

Undefined Item-xx

未定義アイテムxxを使用した。つまり指定されたリロケータブルファイルの内容がおかしい。WZDを使用している場合には起こらない。

Multi Defined

同じラベル名が、2カ所以上で定義されている。このメッセージの後ろに16進2桁の数字が表示されたらそのラベル番号を持つ内部ラベルがエラーであり、文字列が表示されたらその名前の外部ラベルがエラーである。

Undefined Label

未定義ラベルが使用された。内部エラーと外部エラーの区別については、Multi Definedエラーのときと同様である。

Too Many Labels

使用されたラベルの数が多すぎる。

Too Far

相対ジャンプ関係のオブジェクトを作成しよ

うとしたが、目的のアドレスが相対ジャンプで届く範囲にない。

Stack Over

演算用のスタックがオーバーした(16レベル以上のスタックが使用された)。

Stack Empty

演算用のスタックが空なのにその内容を参照するアイテムが使用された。

Illegal ORG to xxxx

PC(Position Counter)を後ろ向きにアドレスxxxxに変更しようとした。以下のオブジェクトファイルの内容は保証されない。

DSEG buffer is Over flowed !!

DSEGエリア用のバッファが足りなくなった(DSEG用のバッファは8Kバイト、これ以上のメモリを用いる場合は、WLK.H内のメモリテーブルを変更するか、ほかのセグメントに割り振ってください)。

リスト1

3000 ED 7B 6A 1F ED 5B 76 1F : CE
3008 13 10 06 01 1A 13 A7 28 : 29
3010 07 FE 20 20 F7 04 18 F4 : 4C
3018 C5 CD 23 40 CD 45 40 21 : 68
3020 00 30 22 46 45 22 44 : 88
3028 22 42 45 22 58 45 22 56 : E0
3030 45 AF 32 3D 45 32 14 : 33
3038 32 28 45 32 3C 45 32 48 : CC
3040 45 32 49 45 DD 21 00 50 : 53
3048 F1 FE 01 20 05 CD 0B 34 : 21
3050 18 11 FD 2A 76 1F FD 23 : 05
3058 FD 23 FD 7E 00 FD 23 FE : B9
3060 20 20 F7 FD 7E 00 A7 20 : 79
3068 06 CD 92 3C CD 0B 34 11 : BE
3070 00 45 FD E5 E1 CD C6 36 : D1
3078 E5 FD E1 21 74 33 CD 77 : CF

SUM: BB 35 3C A3 E1 AA BA 07 SEC0

3080 36 21 78 33 C4 77 36 20 : 93
3088 34 CD EA 36 E5 ED 5B 56 : A4
3090 45 B7 ED 52 E1 DC 03 36 : 81
3098 22 56 45 DD 36 00 01 DD : AE
30A0 23 DD 75 00 DD 23 DD 74 : C6
30A8 00 DD 23 3A 48 45 A7 C2 : 30
30B0 B7 32 3E 01 32 48 45 22 : 09
30B8 42 45 C3 B7 32 21 7C 33 : 03
30C0 CD 77 36 21 80 33 C4 77 : 89
30C8 36 20 3A CD EA 36 E5 ED : 4F
30D0 5B 58 45 B7 ED 52 E1 DC : AB
30D8 53 36 22 58 45 22 5A 45 : 09
30E0 DD 36 00 02 DD 23 DD 75 : 67
30E8 00 DD 23 DD 74 00 DD 73 : 51
30F0 3A 49 45 A7 C2 B7 32 3E : 58
30F8 01 32 49 45 22 44 45 22 : 8E

SUM: B6 DF B5 52 1A 0C 3F 91 AD97

3100 46 45 C3 B7 32 21 84 33 : 0F
3108 CD 77 36 21 87 33 C4 77 : 90
3110 36 C2 A7 31 DD 36 00 04 : E7
3118 DD 23 21 00 45 CD 89 36 : F2
3120 11 00 45 21 D3 33 DC 93 : EC
3128 36 AD E5 D1 21 00 45 CD : FC
3130 AF 5E DD E1 3E 01 11 : D8
3138 00 45 CD BC 40 DA 19 36 : 37
3140 CD 26 34 DD 22 3E 45 2A : D3
3148 36 45 7E 23 FE FF CA 98 : 83
3150 31 FE 0E C2 34 36 5E 23 : BC
3158 56 23 ED 53 45 45 11 8D : DC
3160 45 7E 23 12 13 A7 20 F9 : CB
3168 22 3E 45 CD 14 3C DD 47 : DB
3170 31 CA 47 31 2A 40 45 DD : FF
3178 75 00 DD 23 DD 74 00 DD : A3

SUM: BB 0B A8 DC B2 F1 C1 F7 9559

3180 23 CD 82 42 DA 34 36 2A : 22
3188 B0 45 22 AE 45 3E 02 32 : 7C
3190 53 45 CD 28 37 C3 47 31 : FF
3198 DD 36 00 00 DD 23 DD 36 : 26
31A0 00 00 DD 23 C3 B7 32 21 : CD
31A8 8A 33 CD 77 36 21 8D 33 : 18
31B0 C4 77 36 20 1B 11 28 45 : 2A
31B8 21 00 45 CD AF 36 21 28 : 61
31C0 45 CD 89 36 11 28 45 21 : 70
31C8 D7 33 CD 93 36 C3 B7 32 : 5B
31D0 21 90 33 CD 77 36 21 95 : 14
31D8 33 C4 77 36 20 2B DD 36 : 02
31E0 00 FF 11 14 45 21 00 45 : CF
31E8 CD AF 36 21 14 45 CD 89 : 82
31F0 36 11 14 45 21 DB 33 DC : AB
31F8 93 36 3E 01 32 3C 45 32 : ED

SUM: 78 80 3E E6 80 40 A3 7E EB0A

3200 60 45 AF 32 61 45 A3 B7 : A6
3208 32 21 9A 33 CD 77 36 21 : BB
3210 9D 33 C4 77 36 20 2B DD : 69
3218 36 00 FF 11 14 45 21 00 : C0
3220 45 CD AF 36 21 14 45 CD : 3E
3228 89 36 11 14 45 21 DB 33 : 58
3230 DC 93 36 3E 01 32 3C 45 : 97
3238 32 61 45 AF 32 60 45 C3 : 21
3240 B7 32 21 A4 33 CD 77 36 : 5B
3248 21 A7 33 C4 77 36 20 2A : B6
3250 DD 36 00 FF 11 14 45 21 : 9D
3258 00 45 CD AF 36 21 14 45 : 71
3260 CD 89 36 11 14 45 21 DB : F2
3268 33 DC 93 36 3E 01 32 3C : 85
3270 45 32 61 45 32 60 45 C3 : B7
3278 B7 32 DD 36 00 03 DD 23 : FF

SUM: F2 AD 6F FC 86 C9 4B 80 36A3

3280 21 00 45 CD 89 36 11 00 : 03
3288 45 21 DF 33 CD 93 36 DD : FA
3290 E5 D1 21 00 45 CD AF 36 : CE
3298 E5 DD E1 3E 01 11 00 45 : 38
32A0 CD BC 40 21 00 45 DA FB : 04
32A8 35 2A 80 45 22 AE 45 3E : A7
32B0 02 32 53 45 CD 28 37 DD : D5
32B8 E5 E1 11 00 5F B7 ED 52 : 2C
32C0 38 0F 3A 3C 45 A7 20 09 : D2
32C8 21 AA 33 CD 17 37 C3 6D : 49
32D0 36 FD 7E 00 A7 21 CB 33 : 77
32D8 C4 77 36 28 0E 21 CD 33 : C8
32E0 CD 17 37 FD 2A 76 1F FD : D4
32E8 36 00 00 3A 3C 45 A7 CA : 62
32F0 63 30 CD 4D 3A ED 5B 42 : 6B
32F8 45 2A 56 45 B7 ED 52 21 : 21

SUM: 17 66 F5 E3 5B 2E 27 C6 F212

3300 E3 33 ED 4B 42 45 ED 5B : 1D

3308 56 45 1B C4 5D 33 ED 5B : 52
3310 44 45 2A 58 45 B7 ED 52 : 46
3318 21 EF 33 ED 4B 44 45 ED : F1
3320 5B 58 45 1B C4 5D 33 ED : 54
3328 5B 46 45 2A 5A 45 B7 ED : 53
3330 52 21 FB 33 ED 4B 46 45 : 64
3338 ED 5B 5A 45 1B C4 5D 33 : 56
3340 3A 28 45 A7 28 14 3E 04 : CC
3348 11 28 45 CD A6 41 21 28 : 7B
3350 45 DA FB 35 CD 28 3D CD : 4E
3358 FC 41 C3 FA 1F CD 17 37 : 34
3360 60 69 CD BE 1F 21 07 34 : CF
3368 CD 17 37 62 6B CD BE 1F : 92
3370 CD EE 1F C9 2F 50 3A 00 : 5C
3378 2F 70 3A 00 2F 44 3A 00 : 86

SUM: 48 0F E9 9D F7 F0 85 CA 0045

3380 2F 64 3A 00 2F 53 00 2F : 7E
3388 73 00 2F 4D 00 2F 6D 00 : 8B
3390 2F 4E 3A 50 00 2F 6E 3A : DE
3398 70 00 2F 4E 3A 4A 00 2F : 9A
33A0 6E 3A 64 00 2F 4E 00 2F : 8A
33A8 6E 00 53 6F 72 72 79 20 : AD
33B0 21 20 63 6F 6D 6D 61 6E : BC
33B8 64 73 20 61 72 65 20 74 : C3
33C0 6F 6F 6D 6D 61 6E 79 20 : D3
33C8 21 0D 00 2C 00 3F 45 72 : 50
33D0 72 0D 00 4C 49 42 00 4D : A3
33D8 41 50 00 4F 42 4A 00 52 : BE
33E0 45 4C 00 43 53 45 47 20 : D3
33E8 61 72 65 61 20 3A 00 44 : 37
33F0 53 45 47 20 61 72 65 61 : 98
33F8 20 3A 00 57 53 45 47 20 : B0

SUM: FE 95 D8 79 FC F6 86 DF 0AB1

3400 61 72 65 61 20 3A 00 20 : 13
3408 2D 20 00 3E 2A CD F4 1F : 95
3410 ED 5B 76 1F CD D3 1F 1A : B6
3418 FE 1B CA 6D 36 13 1A A7 : 5A
3420 28 E9 D5 FD E1 C9 DD 05 : 4F
3428 E1 CD 22 37 36 FF FE FF : 39
3430 C8 FE E0 C2 34 36 36 E0 : E8
3438 23 CD 22 37 77 23 CD 22 : D2
3440 37 77 23 CD 22 37 77 23 : C5
3448 A7 20 F8 18 DC 21 BC 35 : 91
3450 CD 17 37 3E 01 32 3D 45 : 0E
3458 CD 45 40 2A 42 45 22 56 : D9
3460 45 2A 44 42 22 58 45 22 : 7B
3468 5A 45 3E 01 11 14 45 CD : 15
3470 A6 41 21 14 45 DA FB 35 : 6B
3478 DD 21 00 50 DD 7E 00 DD : 86

SUM: 07 4D D3 4F A5 A1 22 DA EFBE

3480 23 FE FF CA 52 35 FE 01 : 70
3488 20 21 DD 6E 00 DD 23 DD : 69
3490 66 00 DD 23 ED 5B 56 45 : 49
3498 22 56 45 B7 ED 52 DA 7C : 09
34A0 34 3A 60 45 A7 CA EC 35 : 9F
34A8 C3 7C 34 FE 02 20 24 DD : 94
34B0 6E 00 DD 23 DD 66 00 DD : 8E
34B8 23 ED 5B 58 45 22 5A 45 : C9
34C0 22 58 45 B7 ED 52 DA 7C : 0B
34C8 34 3A 61 45 A7 CA EC 35 : 9F
34D0 C3 7C 34 FE 03 20 2E 21 : E3
34D8 C4 35 CD 17 37 DD E5 E1 : B3
34E0 CD 17 37 CD EE 1F 3E 01 : 34
34E8 DD E5 D1 CD BC 40 DD E5 : 1E
34F0 E1 DA FB 35 2A B0 45 22 : 2C
34F8 AE 45 3E 02 32 53 45 CD : CA

SUM: 69 76 B2 B2 CB A0 39 5B 23D0

3500 28 37 C3 7C 34 FE 04 C2 : 96
3508 7C 34 21 C4 35 CD 17 37 : 9E
3510 DD E5 E1 CD 17 37 CD EE : 79
3518 1F 3E 01 DD E5 D1 CD BC : 7A
3520 40 DD E5 E1 DA FB 35 DD : CA
3528 7E 00 DD 23 A7 20 F8 DD : 1A
3530 6E 00 DD 23 DD 66 00 DD : 8E
3538 23 7C B5 CA 7C 34 CD 82 : 1D
3540 42 2A B0 45 22 AE 45 3E : B4
3548 02 32 53 45 CD 28 37 C3 : BB
3550 2F 35 3A 61 45 A7 CA 9D : 52
3558 35 2A 5E 45 7C B5 CA 9D : 5A
3560 35 ED 5B 56 45 2A 44 45 : CB
3568 37 ED 52 DA 7D 35 2A 44 : 70
3570 45 ED 5B 56 45 B7 ED 52 : 1E
3578 CD EC 35 18 0B 2A 5E 45 : DE

SUM: 15 55 F2 A9 01 FA 78 17 0A9D

3580 7C B5 21 CD 35 C4 17 37 : 66
3588 ED 4B 5E 45 21 00 90 7E : 0A
3590 23 C5 E5 CD C1 42 E1 C1 : 3F
3598 0B 7B B1 20 F2 2A 42 45 : F7
35A0 3A 60 45 A7 20 03 2A 44 : 17
35A8 45 20 02 46 3A 62 45 A7 : 35
35B0 28 03 2A 63 45 22 02 46 : 67
35B8 CD FC 41 C9 50 41 53 53 : 0A
35C0 2D 32 0D 00 4C 69 6E 6B : FA
35C8 69 6E 67 20 00 43 53 45 : 39
35D0 47 20 61 6E 64 20 44 53 : 51
35D8 45 47 20 61 72 65 20 70 : 74
35E0 69 6C 65 64 20 75 70 20 : C3
35E8 21 21 0D 00 7C B5 C8 E5 : 2D
35F0 AF CD C1 42 E1 DA 19 36 : 89
35F8 2B 18 F1 E5 21 0B 36 CD : 48

SUM: 91 37 DE 92 B8 38 3A BA EC7E

3600 17 37 E1 CD 17 37 CD EE : 05
3608 1F 18 62 43 61 6E 20 6E : 39

3610 6F 74 20 6F 70 65 6E 20 : D5
3618 00 21 21 36 CD 17 37 18 : AB
3620 4C 66 69 6C 65 20 41 63 : B0
3628 63 65 73 73 20 45 72 72 : F7
3630 6F 72 0D 00 21 3B 36 CD : 4D
3638 17 37 C9 49 6C 6C 65 67 : 04
3640 61 6C 20 4C 49 42 20 46 : 2A
3648 69 6C 65 20 45 72 72 6F : F2
3650 72 0D 00 21 5A 36 CD 17 : 14
3658 37 C9 49 6C 6C 65 67 61 : 4E
3660 6C 20 4F 52 47 20 45 72 : 4R
3668 72 6F 72 0D 00 3A 3D 45 : 1C
3670 A7 CA FC 41 C3 FA 1F FD : 81
3678 E5 D1 7E A7 28 07 1A BE : E2

SUM: B7 2A 3F 1D 4D D7 61 3C B5CF

3680 C0 13 23 18 F5 D5 FD E1 : B6
3688 C9 7E 23 A7 37 C8 FE 2E : 3C
3698 07 F7 C9 06 0F 1A CD DE : BA
3698 36 38 09 05 28 06 FE 2E : D6
36A0 C8 13 18 F1 FE 2E C9 3E : 16
36A8 2E 12 13 CD AF 36 C9 06 : D4
36B0 14 AF 12 7E 23 CD DE 36 : 57
36B8 38 09 05 28 06 12 13 AF : 48
36C0 12 18 F0 EB 23 C9 06 14 : 0B
36C8 AF 12 7E CD DE 36 D8 05 : FD
36D0 C8 23 12 13 AF 12 18 F2 : DB
36D8 CD 17 37 C3 6D 36 FE 2F : AE
36E0 28 06 FE 2C 28 02 A7 C0 : E9
36E8 37 C9 21 00 00 FD 7E 00 : 9C
36F0 16 30 C9 30 D8 FE 3A 38 : BC
36F8 11 16 37 FE 41 D8 FE 47 : BA

SUM: FD 16 65 16 97 1C 99 BD 228E

3700 38 08 16 57 FE 61 D8 FE : E2
3708 67 D0 29 29 29 29 92 16 : 83
3710 00 5F 1F FD 23 18 D6 7E : 04
3718 A7 C8 23 E5 CD F4 1F E1 : 38
3720 18 F5 E5 CD 6E 42 E1 C9 : 19
3728 CD 6E 42 32 50 45 FE FF : 41
3730 C8 21 28 37 E5 ED 73 65 : F2
3738 45 E6 E0 CA 9D 38 FE 20 : C8
3740 CA C8 38 FE 60 CA 10 39 : 38
3748 3A 50 45 E6 F8 FE 90 CA : 05
3750 EE 39 FE 98 CA 00 3A FE : BF
3758 A0 CA 12 3A FE A8 CA 2B : 51
3760 3A 3A 50 45 E6 F7 26 00 : 94
3768 6F 29 11 73 37 19 7E 23 : 0D
3770 66 6F E9 22 39 5F 39 89 : 3A
3778 39 97 39 A5 39 AD 39 DA : A7

SUM: 12 ED BA 97 06 56 69 72 6A72

3780 39 E7 39 E5 3E E5 3E E5 : 84
3788 3E E5 3E E5 3E E5 3E E5 : 8C
3790 3E E5 3E 73 38 73 38 73 : 2A
3798 38 73 38 73 38 73 38 73 : AC
37A0 38 73 38 73 38 73 38 73 : AC
37A8 38 73 38 73 38 73 38 73 : AC
37B0 38 73 38 73 38 73 38 73 : AC
37B8 38 73 38 73 38 73 38 73 : AC
37C0 38 73 38 73 38 73 38 73 : AC
37C8 38 73 38 73 38 73 38 73 : AC
37D0 38 73 38 2E 3A E5 3E E5 : 53
37D8 3E E5 3E E5 3E E5 3E E5 : 8C
37E0 3E E5 3E E5 3E E5 3E E5 : 8C
37E8 3E E5 3E E5 3E E5 3E E5 : 8C
37F0 3E E5 3E 3D 3A 48 3A 55 : AF
37F8 3A 62 3A 6F 3A 79 3A 82 : B4

SUM: A7 3F A7 EB A4 B7 A8 CD BFF0

3800 3A 90 3A 9E 3A B8 3A C9 : 97
3808 3A D7 3A E4 3A F4 3A 09 : 9B
3810 3B 14 3B 05 3E 3E 3E E5 : B5
3818 3E E5 3E E5 3E E5 3E E5 : 8C
3820 3E E5 3E E5 3E E5 3E E5 : 8C
3828 3E E5 3E E5 3E E5 3E E5 : 8C
3830 3E E5 3E 73 38 1E 3B 32 : 97
3838 3B 3E 3B 43 3B 4B 3B 52 : 0A
3840 3B 60 3B 67 3B 81 3B E5 : 19
3848 3E E5 3E E5 3E E5 3E E5 : 8C
3850 3E E5 3E E5 3E E5 3E E5 : 8C
3858 3E E5 3E E5 3E E5 3E E5 : 8C
3860 3E E5 3E E5 3E E5 3E E5 : 8C
3868 3E 8D 3B 97 3B 9E 3B A5 : 56
3870 3B 73 38 11 85 38 CD E5 : 66
3878 1F 3A 50 45 CD C1 1F CD : 68

SUM: AD 7B D8 B4 9F 55 3C B5 AE0F

3880 EE 1F C3 FA 1F 49 6E 74 : 14
3888 65 72 6E 61 6C 20 45 72 : E9
3890 72 6F 72 29 49 54 45 4D : A2
3898 2D 4E 6F 3A 00 CD 8F 3E : BE
38A0 ED 5B AE 45 19 F5 ED 5B : 81
38A8 B0 45 B7 ED 52 51 38 03 : 07
38B0 22 B0 45 E5 3A 50 45 E6 : B1
38B8 1F 47 CD 5D 3C F1 3A 3D : 24
38C0 45 A7 3E 01 CC AD 3B C9 : A8
38C8 3A 50 45 E6 1F 47 CD 5D : 45
38D0 3C CD 14 3C 30 18 F5 CD : 63
38D8 9A 3E F1 3E 00 CD AD 3B : BB
38E0 3E 01 32 52 45 3A 3D 45 : C4
38E8 A7 C4 2A 3F 18 E1 E5 11 : 00
38F0 00 60 19 7E A7 1E 20 0C : AB
38F8 3E 01 32 52 45 3A 3D 45 : C4

SUM: 48 0D B8 EB 19 CC 54 C7 1FF0

3900 A7 C2 F5 3E 29 11 00 70 : 46
3908 19 5E 23 56 CD F1 3D C9 : B4
3910 3A 50 45 E6 1F 3C 47 C5 : 1C

3918 CD 6E 42 CD 37 3E C1 10 : 90
3920 F6 C9 CD 8F 3E ED 5B AE : 4F
3928 45 19 E5 ED 5B B0 45 B7 : 37
3930 ED 52 E1 38 03 22 B0 45 : 72
3938 E5 11 00 60 19 7E E1 A7 : 75
3940 20 10 3E 01 32 52 45 3A : 72
3948 3D 45 A7 28 05 CD F5 3E : 56
3950 18 09 11 00 70 29 19 7E : 62
3958 23 66 6F CD E9 3D C9 CD : 81
3960 8F 3E ED 5B AE 45 19 E5 : 06
3968 ED 5B B0 45 B7 ED 52 E1 : 14
3970 38 03 22 B0 45 E5 11 00 : 48
3978 60 19 36 01 E1 29 11 00 : CB
SUM: 80 9C 8C A2 1C 7E 1F E8 AC0D

3980 70 19 CD 0D 3E 73 23 72 : A9
3988 C9 CD 19 3E E5 7D CD 37 : 53
3990 3E E1 7C CD 37 3E C9 CD : 73
3998 19 3E E5 7C CD 37 3E E1 : DB
39A0 7D CD 37 3E C9 CD 3E 3E : AC
39A8 7D CD 37 3E C9 CD 9A 3E : 2D
39B0 44 4D CD 19 3E 3A 3D 45 : 71
39B8 A7 CA 37 3E 37 ED 42 38 : 84
39C0 0B 24 25 20 0F 7D B7 F2 : A9
39C8 37 3E 18 08 24 20 65 7D : 5B
39D0 B7 FA 37 3E CD 37 3E C3 : 2B
39D8 19 3F CD 9A 3E E5 CD 8F : 3E
39E0 3E D1 19 CD DE 3D C9 CD : A6
39E8 9A 3E CD E9 3D C9 3A 50 : 16
39F0 45 E6 07 3C 47 C5 CD E6 : B5
39F8 42 CD 37 3E C1 10 F6 C9 : 14
SUM: E6 13 1E 97 8F BA B6 65 E416

3A00 3A 50 45 E6 07 3C 47 C5 : 04
3A08 CD 8F 3E CD DE 3D C1 10 : 53
3A10 F6 C9 3A 50 45 E6 07 3C : B7
3A18 47 C5 CD 8F 3E E5 7C CD : D4
3A20 37 3E E1 7D CD 37 3E C1 : D6
3A28 10 EF C9 C3 EE 39 CD 19 : 98
3A30 3E 7C B5 C8 E5 AF CD 37 : CF
3A38 3E E1 2B 18 F4 CD 11 3E : 72
3A40 CD 1D 3E 19 CD E9 3D C9 : FD
3A48 CD 11 3E CD 1D 3E B7 ED : E8
3A50 52 CD E9 3D C9 CD 11 3E : 2A
3A58 CD 1D 3E CD 69 40 CD E9 : 54
3A60 3D C9 CD 11 3E CD 1D 3E : 4A
3A68 CD 7C 40 CD E9 3D C9 CD : 12
3A70 1D 3E 6C 26 00 CD C9 3D : E0
3A78 C9 CD 1D 3E 26 00 CD E9 : CD
SUM: B0 5F 4D E4 65 3B E2 3B 779C

3A80 3D C9 CD 1D 3E 7C 2F 67 : 40
3A88 7D 2F 6F 23 CD E9 3D C9 : FA
3A90 CD 11 3E CD 1D 3E CD 7C : 8D
3A98 40 EB CD E9 3D C9 CD 1D : D1
3AA0 3E CD 11 3E 44 40 21 01 : 0D
3AA8 00 78 B1 CA E9 3D C5 D5 : B3
3AB0 CD 69 40 D1 C1 0B 18 F1 : 1C
3AB8 CD 11 3E CD 1D 3E CB 3C : 4B
3AC0 CB 1D 1D 20 F9 CD E9 3D : 11
3AC8 C9 CD 11 3E CD 1D 3E 29 : 36
3AD0 1D 20 FC CD E9 3D C9 CD : C2
3AD8 1D 3E 7C 2F 67 7D 2F 67 : 88
3AE0 CD E9 3D C9 CD 11 3E CD : A5
3AE8 1D 3E 7C B2 67 7D B3 6F : 8F
3AF0 CD E9 3D C9 CD 11 3E CD : A5
3AF8 1D 3E 7C B2 67 7D B3 6F : 8F
SUM: 41 49 9F EC EE FF D0 E6 14A7

3B00 CD E9 3D C9 CD 11 3E CD : A5
3B08 1D 3E 7C AA 67 7D AB 6F : 7F
3B10 CD E9 3D C9 CD 1D 3E 7C : 60
3B18 65 6F CD E9 3D C9 CD 6E : CB
3B20 42 06 02 A7 28 07 06 03 : 29
3B28 3D 28 02 06 04 78 32 53 : 6E
3B30 45 C9 CD 19 3E 22 54 45 : ED
3B38 3E 01 32 51 45 C9 AF 32 : B1
3B40 51 45 C9 CD 19 3E 7D CD : CD
3B48 37 3E C9 CD 19 3E CD DE : 0D
3B50 3D C9 CD 19 3E 57 C9 CD : 58
3B58 37 3E E1 7D CD 37 3E C9 : DE
3B60 CD 8F 3E CD E9 3D C9 CD : 23
3B68 9A 3E EB CD 19 3E B7 ED : 8B
3B70 52 C8 DA 3F 3E 5F AF CD : D3
3B78 37 3E E1 2B 7C B5 20 F5 : C7
SUM: 0A D4 EA 70 E7 8B 82 B0 05CD

3B80 C9 CD 1D 3E 22 63 45 3E : F9
3B88 01 32 62 45 C9 CD 6E 42 : 20
3B90 A7 C8 CD F4 1F 18 F6 CD : 2A
3B98 1D 3E CD E9 3F C9 CD 1D : 03
3BA0 3E CD BE 1F C9 CD 1D 3E : D9
3BA8 7D CD C1 1F C9 22 11 3C : 62
3BB0 32 13 3C CD 14 3C D2 1F : 8F
3BB8 3F 20 42 CD 6E 3C 26 00 : 3E
3BC0 6F 29 44 4D 60 69 CD AF : 6E
3BC8 40 78 B1 20 F7 2B 2B ED : C3
3BD0 4B 8B 45 CD 9D 40 2A 8B : 7A
3BD8 45 01 00 8D CD 9D 40 3A : 2A
3BE0 13 3C CD 9D 40 ED 41 11 : 3D
3BE8 3C CD 9D 40 11 8D 45 1A : E3
3BF0 13 F5 CD 98 40 F1 A7 20 : 65
3BF8 F6 22 8B 45 C9 2A 5B 3C : 72
SUM: 51 1F 12 27 78 7E 90 EB 17BB

3C00 23 23 3A 13 3C CD 98 40 : 74
3C08 ED 4B 11 3C CD 9D 40 B7 : E6
3C10 C9 00 00 00 CD 6E 3C 26 : 66
3C18 00 6F 29 22 5B 3C 2A 5B : D6
3C20 3C CD AF 40 78 B1 37 C8 : 20
3C28 ED 43 5B 3C 21 05 00 09 : F6
3C30 11 8D 45 1A A7 28 0A 13 : E9
3C38 47 CD AA 40 B8 28 F4 18 : EA

3C40 DD CD 94 1F A7 20 D7 2A : 25
3C48 5B 3C 23 23 CD AA 40 CD : 61
3C50 AF 40 60 69 B7 C0 3E 01 : 6E
3C58 B7 37 C9 00 00 21 8D 45 : AA
3C60 C5 E5 CD 6E 42 E1 C1 77 : 40
3C68 23 10 F5 36 00 C9 E5 C5 : D1
3C70 21 8D 45 06 00 7E 23 A7 : 41
3C78 28 04 80 47 18 F7 78 C1 : 3B
SUM: 29 4D D4 E3 AE E4 96 55 1054

3C80 E1 C9 21 00 00 01 01 02 : CF
3C88 AF CD 9A 1F ED A1 EA 89 : 36
3C90 3C C9 AF 32 67 45 21 00 : B3
3C98 00 22 68 45 21 00 02 ED : DF
3CA0 5B 8B 45 E5 B7 ED 52 E1 : E7
3CA8 30 14 23 23 CD AA 40 A7 : E8
3CB0 20 05 CD E6 3C 18 E8 23 : 37
3CB8 23 CD 21 3D 18 E1 CD EB : FF
3CC0 1F 2A 68 45 7C B5 C8 CD : BC
3CC8 E9 3F 21 D1 3C 2E CD EB : FF
3CD0 C9 20 55 6E 64 65 66 69 : 44
3CD8 6E 65 64 20 4C 61 62 65 : CB
3CE0 6C 28 73 29 0D 00 3E 2D : A8
3CE8 CD F4 1F CD AF 40 5C 06 : 67
3CF0 0A CD AA 40 A7 28 08 CD : 65
3CF8 F4 1F 10 F5 CD 21 3D 04 : 47
SUM: 10 E8 B6 90 E5 48 44 E4 BAB2

3D00 CD F1 1F 10 FB E3 CD BE : 56
3D08 1F CD F1 1F 2A 68 45 23 : F6
3D10 22 68 45 3A 67 45 3C E6 : D7
3D18 03 32 67 45 CC EB 1F E1 : 98
3D20 C9 CD AA 40 A7 C8 18 F9 : 00
3D28 AF 32 67 45 21 00 02 E5 : 95
3D30 ED 5B 8B 45 B7 ED 52 E1 : EF
3D38 D2 73 3D 23 3E 2D CD 0D : 00
3D40 D4 3D CD AA 40 A7 28 1A : B1
3D48 CD AF 40 C5 CD 82 3D E3 : F0
3D50 29 11 00 70 19 7E 23 66 : CA
3D58 6F CD B7 3D CD 9B 3D E1 : B6
3D60 18 CD 23 23 CD 82 3D 11 : C8
3D68 7D 3D CD AE 3D CD 9B 3D : 17
3D70 C3 2F 3D 3E 0D CD D4 3D : 58
3D78 AF CD D4 3D C9 2A 2A 2A : DA
SUM: 88 F5 5A 03 CD F6 A1 2D A46D

3D80 2A 00 06 0A CD AA 40 A7 : 98
3D88 28 08 CD D4 3D 10 F5 CD : E0
3D90 21 3D 04 3E 20 CD D4 3D : 9E
3D98 10 F9 C9 3A 67 45 3C E6 : DA
3DA0 03 32 67 45 3E 20 C2 D4 : D5
3DA8 3D 3E 0D C3 D4 3D 1A A7 : 1D
3DB0 C8 13 CD D4 3D 18 F7 7C : 44
3DB8 CD BC 3D 7D F5 0F 0F 0F : 65
3DC0 0F CD C5 3D 1F CD CA 3D : A3
3DC8 18 0A E6 0F F6 30 FE 3A : 75
3DD0 D8 C6 07 C9 C5 D5 E5 CD : BA
3DD8 C1 42 E1 D1 C1 C9 E5 7D : A1
3DE0 CD 37 3E 1F 7C CD 37 3E : E1
3DE8 C9 D5 54 5D CD F1 3D D1 : 1B
3DF0 C9 3A 6A 45 FE 11 D2 35 : C8
3DF8 3F E5 D5 26 00 6F 29 11 : C8
SUM: B6 87 82 3E 89 29 28 B3 E215

3E00 6B 45 19 D1 73 23 72 E1 : 83
3E08 3C 32 6A 45 C9 AF 32 52 : 19
3E10 45 E5 CD 1D 3E 54 5D E1 : E4
3E18 C9 AF 32 52 45 3A 6A 45 : 2A
3E20 A7 CA 3A 3F 3D 32 6A 45 : 08
3E28 D5 26 00 6F 29 11 6B 45 : 54
3E30 19 7E 23 66 6F D1 C9 CD : F6
3E38 B5 3E 26 00 6F 3A 0D 45 : 44
3E40 A7 C8 3A 53 45 FE 03 28 : 6A
3E48 0C FE 04 C8 3A 60 45 A7 : 5C
3E50 7D C4 C1 42 C9 3A 61 45 : ED
3E58 A7 C8 E5 2A 5E 45 23 22 : 66
3E60 5E 45 7C 11 00 90 19 D1 : AA
3E68 FE 20 30 02 73 C9 21 77 : 24
3E70 3E CD 17 37 C3 6D 36 44 : 03
3E78 53 45 47 20 62 75 66 66 : A2
SUM: C3 80 F3 8A 41 C6 E8 1D FFA3

3E80 65 72 20 69 73 20 6F 76 : D8
3E88 65 72 20 21 21 0D 00 CD : 13
3E90 6E 42 F5 CD 6E 42 67 F1 : 7A
3E98 6F C9 2A 54 45 3A 51 45 : CB
3EA0 A7 C0 2A 56 45 3A 53 45 : FE
3EA8 FE 02 C8 2A 58 45 FE 03 : 90
3EB0 C8 2A 5A 45 C9 F5 E5 3A : 6E
3EB8 53 45 FE 03 28 0F FE 04 : D0
3EC0 28 12 2A 56 45 23 22 56 : 9A
3EC8 45 18 10 2A 58 45 23 22 : 79
3ED0 58 45 18 07 2A 5A 45 23 : A8
3ED8 22 5A 45 2A 54 45 23 22 : C9
3EE0 54 45 E1 F1 C9 11 4B 3F : CF
3EE8 CD E5 1F 3A 50 45 CD C1 : 2E
3EF0 1F CD EE 1F C9 E5 11 5A : 12
3EF8 3F CD E5 1F E1 CD BE 1F : 9B
SUM: CD AD 13 8D B3 39 EF 35 C56E

3F00 CD EE 1F C9 E5 11 6D 3F : 45
3F08 CD E5 1F E1 CD BE 1F CD : 29
3F10 EE 1F C9 11 84 3F C3 44 : B1
3F18 3F 11 94 3F C3 44 3F 11 : 7A
3F20 9C 3F CD E5 1F 11 8D 45 : 8F
3F28 18 1A 11 B1 3F CD E5 1F : 04
3F30 11 8D 45 18 0F 11 C2 3F : 1C
3F38 18 0A 11 D1 3F 18 05 11 : 71
3F40 DD 3F 18 00 CD E5 1F CD : D2
3F48 EE 1F C9 55 6E 64 65 66 : C8
3F50 69 6E 65 64 20 49 54 45 : A2
3F58 4D 00 55 6E 64 65 66 69 : A8
3F60 6E 65 64 20 4C 61 62 65 : CB

3F68 6C 2D 4E 6F 00 4D 75 6C : 84
3F70 74 69 20 44 65 66 69 6E : E3
3F78 65 64 20 4C 61 62 65 6C : C9
SUM: D8 1E 5C BF 76 C6 AA A1 29F6

3F80 2D 4E 6F 00 54 6F 6F 20 : 3C
3F88 4D 61 6E 79 20 4C 61 62 : C4
3F90 65 6C 73 00 54 6F 6F 20 : 96
3F98 46 61 72 00 4D 75 6C 74 : BB
3FA0 69 20 44 65 66 69 6E 65 : D4
3FA8 64 20 4C 61 62 65 6C 2D : 91
3FB0 00 55 6E 64 65 66 69 6E : C9
3FB8 65 64 20 4C 61 62 65 6C : C9
3FC0 2D 00 53 74 61 63 6B 20 : 43
3FC8 4F 76 65 72 66 6C 6F 77 : 54
3FD0 06 53 74 61 63 6B 20 45 : 5B
3FD8 6D 70 74 79 00 49 6C 6C : EB
3FE0 65 67 61 6C 20 4F 52 47 : A1
3FE8 00 D5 F5 AF 32 4F 45 06 : 45
3FF0 05 11 4F 45 0E 0A CD 13 : A2
3FF8 40 C6 30 1B 12 10 F5 6B : D3
SUM: EA C1 55 2A 3F 70 12 95 F539

4000 62 06 04 7E FE 30 20 05 : 3D
4008 36 20 23 18 F6 CD E5 1F : 58
4010 F1 E1 C9 C5 AF 06 10 29 : 4E
4018 17 2C 91 30 02 2D 81 10 : C4
4020 F6 C1 C9 AF 32 51 45 32 : 29
4028 52 45 32 62 45 21 00 02 : 93
4030 22 8B 45 01 00 10 21 00 : 24
4038 60 36 00 23 0B 78 B1 20 : 0D
4040 F8 CD 82 3C C9 AF 32 6A : 97
4048 45 32 DD 42 3D 32 95 42 : DC
4050 21 00 00 22 B0 45 22 AE : 08
4058 45 22 56 45 22 58 45 22 : 93
4060 5A 45 22 54 45 22 5E 45 : 1F
4068 C9 44 4D 21 00 00 3E 10 : C9
4070 29 CB 23 CB 12 30 01 09 : 2E
4078 3D 20 F5 C9 42 4B 54 5D : 59
SUM: 96 8F FD AE 98 45 CC E8 D220

4080 3E 10 26 00 CB 23 CB 12 : 3F
4088 29 E5 B7 ED 42 E1 38 03 : 10
4090 ED 42 13 3D 20 EE EB C9 : 41
4098 CD 9A 1F 23 C9 F5 79 CD : AD
40A0 9A 1F 23 78 CD 9A 1F 23 : FD
40A8 F1 C9 CD 94 1F 23 C9 F5 : 1B
40B0 CD 94 1F 4F CD CD 94 1F : 72
40B8 47 23 F1 C9 D3 A3 1F D8 : 8B
40C0 CD 25 41 D8 2A 74 1F 11 : D9
40C8 23 46 01 20 00 ED B0 3A : 61
40D0 5D 1F 32 53 46 AF 32 B4 : DC
40D8 45 CD A9 43 D8 06 10 0E : FA
40E0 00 3A 41 46 11 43 46 12 : 6D
40E8 13 FE 7F 30 0F 2A 62 1F : 7A
40F0 85 6F 30 01 24 7E 05 28 : F4
40F8 28 0C 18 EB 0D F5 79 87 : 39
SUM: 12 7A 34 61 6B 0A 39 A7 3C0E

4100 87 87 87 AF F1 D6 80 81 : AC
4108 32 59 46 AF 32 54 46 01 : CD
4110 37 00 11 B5 45 21 23 46 : 9A
4118 ED B0 3E FF 32 95 42 B7 : CC
4120 C9 3E 07 37 C9 3A 5D 1F : C4
4128 CD 44 41 D8 CD 4F 1D 08 : 5F
4130 3E 08 37 C0 E5 ED 5B 74 : 3C
4138 1F 01 20 00 ED B0 E1 7E : 0E
4140 CD 99 41 C9 FE 41 38 04 : EB
4148 FE 45 3F D0 3E 03 C9 0E : 6A
4150 10 ED 5B 60 1F ED 53 55 : 6C
4158 46 2A 64 1F 3E 1D CD 00 : FF
4160 20 D8 06 08 22 57 46 7E : 43
4168 FE FF 28 1A B7 28 0B D5 : FE
4170 ED 5B 74 1F CD 8A 41 D1 : 44
4178 28 0D D5 11 20 00 19 D1 : 25
SUM: 24 4F 71 EB 61 41 D1 C4 5EAB

4180 10 E2 13 0D 20 CF 3E AF : EE
4188 B7 C9 C5 E5 06 10 13 23 : 76
4190 1A BE 20 02 10 F8 E1 C1 : A4
4198 C9 E5 E6 87 21 1F 29 BE : 5D
41A0 E1 C8 3E 06 37 C9 CD A3 : 52
41A8 1F D8 CD AF 1F D8 01 20 : 8B
41B0 00 11 23 46 2A 74 1F ED : 24
41B8 B0 3A 5D 1F 32 53 46 2A : 5B
41C0 E1 27 22 57 46 2A DF 27 : F7
41C8 22 55 46 CD D8 43 D8 32 : AF
41D0 41 46 32 43 46 3E 80 32 : 32
41D8 44 46 AF 32 59 46 32 B4 : F0
41E0 45 32 54 46 21 00 00 22 : 54
41E8 35 46 01 37 00 11 EC 45 : F5
41F0 21 23 46 ED B0 3E 00 32 : 97
41F8 DD 42 B7 C9 21 EC 45 11 : 02
SUM: 5A 1E 04 61 B8 8A 28 14 B113

4200 23 46 01 37 00 ED B0 3A : 78
4208 53 46 32 5D 1F 2A 35 46 : EC
4210 2C 2D C4 F0 42 3A 59 46 : 28
4218 2A 35 46 2C 2D 20 01 3C : 5B
4220 67 22 35 46 3E 01 ED 5B : 88
4228 55 46 2A 64 1F CD 00 20 : 35
4230 D8 21 23 46 ED 5B 57 46 : 47
4238 01 20 00 ED B0 3E 01 ED : EA
4240 5B 55 46 2A 64 1F CD 03 : 73
4248 20 D8 CD A9 43 D8 06 10 : 9F
4250 21 43 46 7E FE 7F 30 12 : E7
4258 23 4E E5 2A 62 1F 16 00 : 17
4260 5F 19 71 E1 05 CA 21 41 : FB
4268 18 E9 CD C7 43 C9 21 95 : 57
4270 42 34 20 08 CD 97 42 D8 : 1C
4278 21 E6 45 34 2A 95 42 7E : FF
SUM: FA 71 A0 EC CE 2C 63 01 A6F9

▶この前、電器屋さんでMZ-2200の中古がデータレコーダとセットで8,000円で、売られていました。別のところでは初代PC-8001が本体のみ2,500円で売られてました。時の流れは速い。
本間 晃(19)愛知県


```

4280 B7 C9 7D 3D 32 95 42 7C : BF
4288 32 E6 45 CD 97 42 D8 21 : FC
4290 E6 45 34 B7 C9 00 B0 01 : 90
4298 37 00 11 23 46 21 B5 45 : CC
42A0 ED B0 3A 53 46 32 5D 1F : 1E
42A8 3A 54 46 47 3A 59 46 B8 : AC
42B0 DA 21 41 78 CD 8D 43 EB : 3C
42B8 3E 01 21 00 B0 CD 00 20 : FD
42C0 C9 2A FE 45 23 22 FE 45 : BE
42C8 2A DD 42 77 21 DD 42 34 : 34
42D0 37 3F C0 CD DF 42 D8 21 : 1D
42D8 1D 46 34 B7 C9 00 B1 01 : C9
42E0 37 00 11 23 46 21 EC 45 : 03
42E8 ED B0 3A 53 46 32 5D 1F : 1E
42F0 3A 54 46 47 3A 59 46 B8 : AC
42F8 30 76 CD A9 43 D8 3A 54 : C5

```

SUM: 1A 20 7B 9C CA A2 F7 D0 5907

```

4300 46 E6 F0 47 3A 59 46 B8 : 22
4308 F0 B8 30 47 CB 3F CB 3F : 33

```

```

4310 CB 3F CB 3F 21 44 46 16 : D5
4318 00 5F 19 E5 3A 59 46 E6 : 1C
4320 F0 CD 8D 43 CD FA 43 2A : C1
4328 62 1F 16 00 5F 19 36 8F : D4
4330 CD D8 43 77 E1 D8 77 23 : B2
4338 36 80 2A 62 1F 16 00 5F : D6
4340 19 36 80 3A 59 46 E6 F0 : 7C
4348 C6 10 32 59 46 CD C7 43 : 7E
4350 D8 18 9D 3A 54 46 32 59 : EC
4358 46 CB 3F CB 3F CB 3F CB : 2F
4360 3F 21 44 46 16 00 5F 19 : 28
4368 3A 54 46 E6 0F C6 80 77 : 86
4370 3A 54 46 CD 8D 43 EB 3E : 7A
4378 01 21 00 B1 CD 03 20 D8 : 9B

```

SUM: 07 93 72 10 3D 66 95 59 09A9

```

4380 01 37 00 11 EC 45 21 23 : BE
4388 46 ED B0 B7 C9 F5 F5 CB : 18
4390 3F CB 3F CB 3F CB 3F 21 : 7E

```

```

4398 43 46 16 00 5F 19 7E CD : 62
43A0 F2 43 F1 E6 0F 85 6F F1 : 00
43A8 C9 D5 E5 3A B4 45 57 3A : 47
43B0 5D 1F BA 28 0F 32 B4 45 : 98
43B8 3E 01 ED 5B 5E 1F 2A 62 : 90
43C0 1F CD 00 20 E1 D1 C9 D5 : 5C
43C8 E5 3E 01 ED 5B 5E 1F 2A : 13
43D0 62 1F CD 03 20 E1 D1 C9 : EC
43D8 C5 E5 06 80 2A 62 1F 7E : 59
43E0 B7 28 08 23 10 F9 3E 09 : 5A
43E8 37 18 04 3E 80 90 A7 E1 : 29
43F0 C1 C9 26 00 6F 29 29 29 : 9A
43F8 29 C9 E5 CB 3C CB 1D CB : 91

```

SUM: 22 4E 6D F2 44 28 7A D2 ABAF

```

4400 3C CB 1D CB 3C CB 1D CB : DE
4408 3C CB 1D 7D E1 C9 : 4B

```

SUM: 78 96 3A 48 1D 94 1D CB CF2A

リスト2

```

1 38000* ;
2 0000* ;
3 0000* ; Wk Link Program Ver 1.00
4 0000* ; Programmed by T. Ishigami
5 0000* ; '90 Mar 7th
6 0000* ;
7 0000* ; CSEG 3000H -
8 0000* ; DSEG 4500H -
9 0000* ;
10 0000* ;
11 0000* ;
12 0000* ;
13 0000* ; CSEG
14 0000* ;
15 0007 NAME EQU 15
16 0003 EXT EQU 3
17 0014 LEN EQU NAME + 1 + EXT + 1
18 0000* ;
19 0000 CR EQU 0DH
20 0018 BKK EQU 1BH
21 00F5 EOF EQU -1
22 0000* ;
23 0001 PRG EQU 1
24 0002 DTA EQU 2
25 0003 REL EQU 3
26 0004 LIB EQU 4
27 0005 MAP EQU 5
28 0000* ;
29 0000* C INCLUDE SOS.DEF
30 1000 C NOT EQU 1FFAH
31 1004 C PRINT EQU 1FAFH
32 10F1 C PRINTS EQU 1FF3H
33 10FE C LTRN EQU 1FE2H
34 10FB C NL EQU 1FB2H
35 10F5 C NCR EQU 1F52H
36 10FD C TAB EQU 1FDFH
37 10D3 C GETL EQU 1D33H
38 10C7 C PAUSE EQU 1C77H
39 10C1 C PTRX EQU 1FC1H
40 10BE C PTRHL EQU 1FB2H
41 10B6 C HRC EQU 1B6H
42 10A3 C FILE EQU 1FA3H
43 10AF C WOPEN EQU 1FAFH
44 10FA C POKE EQU 1FAAH
45 10F4 C PEEX EQU 1FA4H
46 2009 C HOPEN EQU 2009H
47 2015 C KILL EQU 2015H
48 2012 C NAME EQU 2012H
49 20C3 C EXOR EQU 20C3H
50 0000* ;
51 2000 C DORS EQU 2000H
52 2003 C DORS EQU 2003H
53 0000* ;
54 107A C PRONT EQU 107AH
55 1076 C KRFAD EQU 1076H
56 1074 C KRFAD EQU 1074H
57 1072 C SIZE EQU 1072H
58 10FA C KEMAL EQU 1FAAH
59 10FA C JTOP EQU 1FAAH
60 10F2 C FATP EQU 1F22H
61 10F0 C DIRPS EQU 1F00H
62 10F0 C FATPQS EQU 1F02H
63 10F0 C DSK EQU 1F00H
64 0000* ;
65 0000* C INCLUDE WIK.DEF
66 0000* C Header File For Wk
67 0000* C ; CSEG 3000H -
68 0000* C ; DSEG 4500H -
69 0000* ;
70 1000 C LRLMAX EQU 1000H
71 0000* ;
72 5000 C Cmbuf EQU 5000H ;= 5FFFH
73 5704 C Cmbint EQU 5704H
74 0000* ;
75 0000 C LRLFLG EQU 0000H ;= 0FFFH
76 7000 C LRLJUM EQU 7000H ;= 0FFFH
77 0000* ;
78 0000 C RF_DSEG EQU 0000H ;= 0FFFH
79 0000 C RDRIF EQU 0000H ;= 0FFFH
80 0100 C RDRIF EQU 0100H ;= 0FFFH
81 0000* ;
82 0000* ;
83 0000* ; External
84 0000* ;
85 EXT LRLOFF
86 EXT NYTOFF
87 EXT FLGCD
88 EXT FLGDT
89 EXT FLGEX
90 EXT EXADR
91 EXT PTRCD
92 EXT PRDPT
93 EXT PTRNE
94 EXT CNTDT
95 EXT WKOUT
96 EXT SEGMD
97 0000* ;
98 0000* ; Start
99 0000* ;
100 0000* ED 7B 6A 1F LD SP, (HEMAL)
101 0004* ;
102 0004* ED 5B 76 1F LD DE, (KRFAD)
103 0004* 13 INC DE
104 0007* 13 INC DE ;Skip '0' or 'B'
105 000A* 00 01 LD B, 1 ;B = argc
106 000C* 1A CCL: LD A, (DE)
107 000D* 13 INC DE
108 000E* A7 AND A
109 000F* 28 07 JR Z, CC3
110 0011* FE 28 JR NZ, CC1
111 0013* 28 F7 JR NZ, CC1
112 0015* 04 INC B
113 0016* 18 F4 JK CC1 ;if (DE) = "" then arg
114 0010* ;
115 0010* C5 CCL: PUSH BC
116 0010* 13 INC DE
117 0019* C0 ** CALL INIWK1
118 001C* C0 ** CALL INIWK2
119 0017* 21 00 LD HL, 3000H
120 0022* 22 ** LD (strw4), HL

```

```

121 0025* 22 ** LD (strdt), HL
122 0028* 22 ** LD (strcd), HL
123 0028* 22 ** LD (PTRDT), HL
124 002C* 22 ** LD (PTRCD), HL
125 0031* AF XOR A
126 0032* 32 ** LD (pass), A
127 0035* 32 ** LD (obname), A
128 0036* 32 ** LD (aname), A
129 0038* 32 ** LD (endf), A
130 0035* 32 ** LD (sfgcd), A
131 0041* 32 ** LD (sfgcd), A
132 0044* D0 21 00 50 LD IX, Cmbuf ;IX means Command p
133 0040* ;
134 0040* F1 POP AF ;A = argc
135 0049* FE 01 CP 1
136 0048* 28 05 JR NZ, CC8
137 0040* C0 ** CALL GETL
138 0050* 18 11 JR CC7
139 0052* ;
140 0052* FD 2A 76 1F CCL: LD IV, (KRFAD)
141 0050* FD 23 INC IV
142 0050* FD 23 INC IV ;Skip '0' or 'B'
143 005A* FD 7E 00 CCL: LD A, (IV)
144 0050* FD 23 INC IV
145 005F* FE 28 CP ''
146 0061* 28 F7 JR NZ, CC4
147 0063* ;
148 0063* FD 7E 00 CCL: LD A, (IV)
149 0066* A7 AND A
150 0067* 28 06 JR NZ, CC8
151 0066* C0 ** CALL PTRAP
152 006C* C0 ** CALL GETL
153 006F* ;
154 006F* ; CCL:
155 006F* 11 ** LD DE, renase
156 0072* FD E5 PUSH HL
157 0074* E1 POP HL ;LD HL, IV
158 0075* C0 ** CALL copy2
159 0078* F5 PUSH HL
160 0079* FD E1 POP IV ;LD IV, HL
161 0079* ;
162 0079* 21 ** LD HL, swP
163 007E* C0 ** LD HL, swP4
164 0081* 21 ** CALL NZ, swatch
165 0084* C4 ** JR NZ, CC9
166 0087* 28 34 JR NZ, CC9
167 0089* ;
168 0089* C0 ** CALL hhex
169 008C* E5 PUSH HL
170 0090* ED 58 ** LD DE, (PTRCD)
171 0091* 07 OR A
172 0092* ED 52 SRC HL, DE
173 0094* E1 POP HL
174 0095* C0 ** CALL EXR14 ;illegal ORG Error
175 0095* ;
176 0098* 22 ** LD (PTRCD), HL
177 0099* ;
178 0099* 00 30 00 01 LD (IX), PRG
179 0099* 00 23 INC IX
180 00A1* 00 75 00 LD (IX), I
181 00A4* 00 74 00 INC IX
182 00A5* 00 74 00 LD (IX), I
183 00A9* 00 23 INC IX
184 00AB* ;
185 00AB* 3A ** LD A, (sfgcd)
186 00AE* A7 AND A
187 00AF* C2 ** JP NZ, CC12
188 00B2* ;
189 00B2* 3E 01 LD A, 1
190 00B4* 32 ** LD (sfgcd), A
191 00B7* 22 ** LD (strcd), HL
192 00B4* C3 ** JP CC12
193 00B7* ;
194 00B0* 21 ** CCL: LD HL, sw0
195 00C0* C0 ** CALL amatch
196 00C3* 21 ** LD HL, swP4
197 00C0* C4 ** CALL NZ, swatch
198 00C3* 28 3A JR NZ, CC13
199 00C5* ;
200 00C8* C0 ** CALL hhex
201 00CE* E5 PUSH HL
202 00CF* ED 58 ** LD DE, (PTRDT)
203 00D3* 07 OR A
204 00D4* ED 52 SRC HL, DE
205 00D6* E1 POP HL ;illegal ORG Error
206 00D7* 0C ** CALL C, RDR14
207 00D8* ;
208 00D8* 22 ** LD (PTRDT), HL
209 00D9* 22 ** LD (PTRNE), HL
210 00E0* ;
211 00E0* 00 30 00 02 LD (IX), DTA
212 00E4* 00 23 INC IX
213 00E5* 00 75 00 LD (IX), I
214 00E9* 00 23 INC IX
215 00E9* 00 74 00 LD (IX), H
216 00E2* 00 23 INC IX
217 00F0* ;
218 00F0* 3A ** LD A, (sfgcd)
219 00F3* A7 AND A
220 00F4* C2 ** JP NZ, CC12
221 00F7* 3E 01 LD A, 1
222 00F9* 32 ** LD (sfgcd), A
223 00FC* 22 ** LD (strw4), HL
224 00FF* 22 ** LD (strw4), HL
225 0102* C3 ** JP CC12
226 0105* ;
227 0105* 21 ** CCL: LD HL, sw5
228 0108* C0 ** CALL amatch
229 010B* 21 ** LD HL, swP3
230 010E* C4 ** CALL NZ, swatch
231 0111* C2 ** JP NZ, CC17
232 0114* ;
233 0114* 00 36 00 04 LD (IX), LIB
234 0118* 00 20 INC IX
235 011A* 21 ** LD HL, renase
236 011D* C0 ** CALL there
237 0120* ;
238 0120* 11 ** LD DE, renase
239 0123* 21 ** LD HL, strLIB
240 0126* C0 ** CALL C, flineat

```

```

241 0129* ;
242 0129* 00 E5 PUSH IX ;LD DE, IX
243 0128* 01 POP DE
244 012C* 21 ** LD HL, renase
245 012F* C0 ** CALL fcopy
246 0132* F5 PUSH HL ;LD IX, HL
247 0130* D0 E1 POP IX
248 0135* ;
249 0135* 3E 01 LD A, 1 ;Bin File
250 0137* 11 ** LD DE, renase
251 013A* C0 ** CALL RDRP2N
252 013D* DA ** JP C, ferr
253 0140* ;
254 0140* C0 ** CALL getwad (Hseek), IX
255 0143* D0 22 ** LD HL, (Hseek), IX
256 0147* ;
257 0147* 2A ** CCL: LD HL, (Hseek)
258 014A* 7E LD A, (HL)
259 014B* 23 INC HL
260 014C* ;
261 014C* FE FF CP EOF
262 014E* CA ** JP Z, CC16
263 0151* FE FF CP BEH
264 0153* C2 ** JP NZ, ferr
265 0158* ;
266 0158* 5E LD E, (HL)
267 0157* 23 INC HL
268 0158* 58 LD D, (HL)
269 0159* 23 INC HL
270 015A* ED 53 ** LD (Hseek), DE
271 015E* ;
272 015E* 11 ** LD DE, LRLBUF
273 0161* 7E LD A, (HL)
274 0162* 23 INC HL
275 0163* 12 LD (DE), A
276 0164* 13 INC DE
277 0165* A7 AND A
278 0166* 28 F9 JR NZ, CC15
279 0168* 22 ** LD (Hseek), HL
280 0168* ;
281 0168* C0 ** CALL SEALBL
282 016E* 02 ** JP Z, CC14
283 0171* CA ** LD HL, swP
284 0174* ;
285 0174* 2A ** LD (IX), I
286 0177* D0 75 00 LD (IX), I
287 017A* D0 23 INC IX
288 017C* D0 74 00 LD (IX), H
289 017F* 00 23 INC IX
290 0181* ;
291 0181* C0 ** CALL PSEAL
292 0184* DA ** JP C, ferr
293 0187* ;
294 0187* 2A ** LD HL, (XTOFF), HL
295 018A* 22 ** LD (LRLOFF), HL
296 018B* 3E 02 LD A, 2 ;CSEG
297 018F* 32 ** LD (SEGMD), A
298 0192* C0 ** CALL asm
299 0195* C3 ** JP CC14
300 0195* ;
301 0198* D0 36 00 00 CCL: LD (IX), 0
302 019C* D0 23 INC IX
303 019E* D0 36 00 00 LD (IX), 0
304 01A2* D0 23 INC IX
305 01A4* C3 ** JP CC12
306 01A7* ;
307 01A7* 21 ** CCL: LD HL, swM
308 01AA* C0 ** CALL amatch
309 01AD* 21 ** LD HL, swP3
310 01B0* C4 ** CALL NZ, swatch
311 01B3* 28 1B JR NZ, CC22
312 01B5* ;
313 01B5* 11 ** LD DE, renase
314 01B8* 21 ** LD HL, renase
315 01B9* C0 ** CALL fcopy
316 01BE* 21 ** LD HL, renase
317 01C1* C0 ** CALL there
318 01C4* ;
319 01C4* 11 ** LD DE, renase
320 01C7* 21 ** LD HL, strMAP
321 01CA* DC ** CALL C, flineat
322 01C9* C3 ** JP CC12
323 01D0* ;
324 01D0* 21 ** CCL: LD HL, swNP
325 01D3* C0 ** CALL amatch
326 01D6* 21 ** LD HL, swNP+5
327 01D9* C4 ** CALL NZ, swatch
328 01DC* 28 28 JR NZ, CC27
329 01DE* ;
330 01DE* 00 36 00 FF LD (IX), EOF
331 01E2* 11 ** LD DE, obname
332 01E5* 21 ** LD HL, renase
333 01E8* C0 ** CALL fcopy
334 01EB* 21 ** LD HL, obname
335 01EE* C0 ** CALL there
336 01F1* 11 ** LD DE, obname
337 01F4* 21 ** LD HL, strOBJ
338 01F7* DC ** CALL C, flineat
339 01FA* ;
340 01FA* 3E 01 LD A, 1
341 01FC* 32 ** LD (endf), A
342 01FF* 32 ** LD (FLGCD), A
343 0202* AF XOR A
344 0205* 32 ** LD (FLGDT), A
345 0206* C3 ** JP CC12
346 0209* ;
347 0209* 21 ** CCL: LD HL, swNP
348 020C* C0 ** CALL amatch
349 020F* 21 ** LD HL, swNP+3
350 0212* C4 ** CALL NZ, swatch
351 0215* 28 20 JR NZ, CC32
352 0217* ;
353 0217* 00 36 00 FF LD (IX), EOF
354 021B* 11 ** LD DE, obname
355 021E* 21 ** LD HL, renase
356 0221* C0 ** CALL fcopy
357 0224* 21 ** LD HL, obname
358 0227* C0 ** CALL there
359 022A* 11 ** LD DE, obname
360 022D* 21 ** LD HL, strOBJ
361 0230* DC ** CALL C, flineat

```

```

362 0233* 3E 01 LD A,1
363 0233* 3E 01 LD (endf1g),A
364 0235* 31 00 LD (FLGDT),A
365 0236* 32 00 LD (FLGDT),A
366 0238* 32 00 LD (FLGDT),A
367 0239* 32 00 LD (FLGDT),A
368 0239* 32 00 LD (FLGDT),A
369 0242* 31 00 LD (FLGDT),A
370 0242* 31 00 LD (FLGDT),A
371 0245* 31 00 LD (FLGDT),A
372 0246* 31 00 LD (FLGDT),A
373 0248* 31 00 LD (FLGDT),A
374 0248* 31 00 LD (FLGDT),A
375 0250* 31 00 LD (FLGDT),A
376 0250* 31 00 LD (FLGDT),A
377 0251* 31 00 LD (FLGDT),A
378 0251* 31 00 LD (FLGDT),A
379 0252* 31 00 LD (FLGDT),A
380 0252* 31 00 LD (FLGDT),A
381 0253* 31 00 LD (FLGDT),A
382 0253* 31 00 LD (FLGDT),A
383 0254* 31 00 LD (FLGDT),A
384 0254* 31 00 LD (FLGDT),A
385 0255* 31 00 LD (FLGDT),A
386 0255* 31 00 LD (FLGDT),A
387 0256* 31 00 LD (FLGDT),A
388 0256* 31 00 LD (FLGDT),A
389 0257* 31 00 LD (FLGDT),A
390 0257* 31 00 LD (FLGDT),A
391 0258* 31 00 LD (FLGDT),A
392 0258* 31 00 LD (FLGDT),A
393 0259* 31 00 LD (FLGDT),A
394 0259* 31 00 LD (FLGDT),A
395 0260* 31 00 LD (FLGDT),A
396 0260* 31 00 LD (FLGDT),A
397 0261* 31 00 LD (FLGDT),A
398 0261* 31 00 LD (FLGDT),A
399 0262* 31 00 LD (FLGDT),A
400 0262* 31 00 LD (FLGDT),A
401 0263* 31 00 LD (FLGDT),A
402 0263* 31 00 LD (FLGDT),A
403 0264* 31 00 LD (FLGDT),A
404 0264* 31 00 LD (FLGDT),A
405 0265* 31 00 LD (FLGDT),A
406 0265* 31 00 LD (FLGDT),A
407 0266* 31 00 LD (FLGDT),A
408 0266* 31 00 LD (FLGDT),A
409 0267* 31 00 LD (FLGDT),A
410 0267* 31 00 LD (FLGDT),A
411 0268* 31 00 LD (FLGDT),A
412 0268* 31 00 LD (FLGDT),A
413 0269* 31 00 LD (FLGDT),A
414 0269* 31 00 LD (FLGDT),A
415 0270* 31 00 LD (FLGDT),A
416 0270* 31 00 LD (FLGDT),A
417 0271* 31 00 LD (FLGDT),A
418 0271* 31 00 LD (FLGDT),A
419 0272* 31 00 LD (FLGDT),A
420 0272* 31 00 LD (FLGDT),A
421 0273* 31 00 LD (FLGDT),A
422 0273* 31 00 LD (FLGDT),A
423 0274* 31 00 LD (FLGDT),A
424 0274* 31 00 LD (FLGDT),A
425 0275* 31 00 LD (FLGDT),A
426 0275* 31 00 LD (FLGDT),A
427 0276* 31 00 LD (FLGDT),A
428 0276* 31 00 LD (FLGDT),A
429 0277* 31 00 LD (FLGDT),A
430 0277* 31 00 LD (FLGDT),A
431 0278* 31 00 LD (FLGDT),A
432 0278* 31 00 LD (FLGDT),A
433 0279* 31 00 LD (FLGDT),A
434 0279* 31 00 LD (FLGDT),A
435 0280* 31 00 LD (FLGDT),A
436 0280* 31 00 LD (FLGDT),A
437 0281* 31 00 LD (FLGDT),A
438 0281* 31 00 LD (FLGDT),A
439 0282* 31 00 LD (FLGDT),A
440 0282* 31 00 LD (FLGDT),A
441 0283* 31 00 LD (FLGDT),A
442 0283* 31 00 LD (FLGDT),A
443 0284* 31 00 LD (FLGDT),A
444 0284* 31 00 LD (FLGDT),A
445 0285* 31 00 LD (FLGDT),A
446 0285* 31 00 LD (FLGDT),A
447 0286* 31 00 LD (FLGDT),A
448 0286* 31 00 LD (FLGDT),A
449 0287* 31 00 LD (FLGDT),A
450 0287* 31 00 LD (FLGDT),A
451 0288* 31 00 LD (FLGDT),A
452 0288* 31 00 LD (FLGDT),A
453 0289* 31 00 LD (FLGDT),A
454 0289* 31 00 LD (FLGDT),A
455 0290* 31 00 LD (FLGDT),A
456 0290* 31 00 LD (FLGDT),A
457 0291* 31 00 LD (FLGDT),A
458 0291* 31 00 LD (FLGDT),A
459 0292* 31 00 LD (FLGDT),A
460 0292* 31 00 LD (FLGDT),A
461 0293* 31 00 LD (FLGDT),A
462 0293* 31 00 LD (FLGDT),A
463 0294* 31 00 LD (FLGDT),A
464 0294* 31 00 LD (FLGDT),A
465 0295* 31 00 LD (FLGDT),A
466 0295* 31 00 LD (FLGDT),A
467 0296* 31 00 LD (FLGDT),A
468 0296* 31 00 LD (FLGDT),A
469 0297* 31 00 LD (FLGDT),A
470 0297* 31 00 LD (FLGDT),A
471 0298* 31 00 LD (FLGDT),A
472 0298* 31 00 LD (FLGDT),A
473 0299* 31 00 LD (FLGDT),A
474 0299* 31 00 LD (FLGDT),A
475 0300* 31 00 LD (FLGDT),A
476 0300* 31 00 LD (FLGDT),A
477 0301* 31 00 LD (FLGDT),A
478 0301* 31 00 LD (FLGDT),A
479 0302* 31 00 LD (FLGDT),A
480 0302* 31 00 LD (FLGDT),A
481 0303* 31 00 LD (FLGDT),A
482 0303* 31 00 LD (FLGDT),A
483 0304* 31 00 LD (FLGDT),A
484 0304* 31 00 LD (FLGDT),A
485 0305* 31 00 LD (FLGDT),A
486 0305* 31 00 LD (FLGDT),A
487 0306* 31 00 LD (FLGDT),A
488 0306* 31 00 LD (FLGDT),A
489 0307* 31 00 LD (FLGDT),A
490 0307* 31 00 LD (FLGDT),A
491 0308* 31 00 LD (FLGDT),A
492 0308* 31 00 LD (FLGDT),A
493 0309* 31 00 LD (FLGDT),A
494 0309* 31 00 LD (FLGDT),A
495 0310* 31 00 LD (FLGDT),A
496 0310* 31 00 LD (FLGDT),A
497 0311* 31 00 LD (FLGDT),A
498 0311* 31 00 LD (FLGDT),A
499 0312* 31 00 LD (FLGDT),A
500 0312* 31 00 LD (FLGDT),A
501 0313* 31 00 LD (FLGDT),A
502 0313* 31 00 LD (FLGDT),A
503 0314* 31 00 LD (FLGDT),A
504 0314* 31 00 LD (FLGDT),A
505 0315* 31 00 LD (FLGDT),A
506 0315* 31 00 LD (FLGDT),A
507 0316* 31 00 LD (FLGDT),A
508 0316* 31 00 LD (FLGDT),A
509 0317* 31 00 LD (FLGDT),A
510 0317* 31 00 LD (FLGDT),A
511 0318* 31 00 LD (FLGDT),A
512 0318* 31 00 LD (FLGDT),A
513 0319* 31 00 LD (FLGDT),A
514 0319* 31 00 LD (FLGDT),A
515 0320* 31 00 LD (FLGDT),A
516 0320* 31 00 LD (FLGDT),A
517 0321* 31 00 LD (FLGDT),A
518 0321* 31 00 LD (FLGDT),A
519 0322* 31 00 LD (FLGDT),A
520 0322* 31 00 LD (FLGDT),A
521 0323* 31 00 LD (FLGDT),A
522 0323* 31 00 LD (FLGDT),A
523 0324* 31 00 LD (FLGDT),A
524 0324* 31 00 LD (FLGDT),A

```



```

846 4628' 41 63 65 73
847 4629' 73 24 45 72 73
848 4630' 67 72 80 80
849 4631' 21 *** fErr: LD HL, C055
850 4632' CD *** CALL puts
851 4633' CD *** RET
852 4634' CD *** DB 'Illegal LIB File Error', CR, 0
853 4635' 49 6C 8C 65 67
854 4636' 01 8C 24 4C 49
855 4637' 41 8C 24 4F 52
856 4638' 45 24 45 72 72
857 4639' 67 72 80 80
858 4640' 21 *** ERR14: LD HL, STR14
859 4641' CD *** CALL puts
860 4642' CD *** RET
861 4643' CD *** DB 'Illegal ORG Error', CR, 0
862 4644' 49 6C 8C 65 67
863 4645' 41 8C 24 4F 52
864 4646' 45 24 45 72 72
865 4647' 67 72 80 80
866 4648' 21 *** ext1: LD A, (pass)
867 4649' CD *** AND A
868 4650' CD *** CALL NZ, CLOSE
869 4651' CD *** JP, HOT
870 4652' 21 *** fErr: LD HL, C055
871 4653' CD *** CALL puts
872 4654' CD *** RET
873 4655' CD *** DB 'Illegal ORG Error', CR, 0
874 4656' 49 6C 8C 65 67
875 4657' 41 8C 24 4F 52
876 4658' 45 24 45 72 72
877 4659' 67 72 80 80
878 4660' 21 *** ext1: LD A, (pass)
879 4661' CD *** AND A
880 4662' CD *** CALL NZ, CLOSE
881 4663' CD *** JP, HOT
882 4664' 21 *** fErr: LD HL, C055
883 4665' CD *** CALL puts
884 4666' CD *** RET
885 4667' CD *** DB 'Illegal ORG Error', CR, 0
886 4668' 49 6C 8C 65 67
887 4669' 41 8C 24 4F 52
888 4670' 45 24 45 72 72
889 4671' 67 72 80 80
890 4672' 21 *** ext1: LD A, (pass)
891 4673' CD *** AND A
892 4674' CD *** CALL NZ, CLOSE
893 4675' CD *** JP, HOT
894 4676' 21 *** fErr: LD HL, C055
895 4677' CD *** CALL puts
896 4678' CD *** RET
897 4679' CD *** DB 'Illegal ORG Error', CR, 0
898 4680' 49 6C 8C 65 67
899 4681' 41 8C 24 4F 52
900 4682' 45 24 45 72 72
901 4683' 67 72 80 80
902 4684' 21 *** ext1: LD A, (pass)
903 4685' CD *** AND A
904 4686' CD *** CALL NZ, CLOSE
905 4687' CD *** JP, HOT
906 4688' 21 *** fErr: LD HL, C055
907 4689' CD *** CALL puts
908 4690' CD *** RET
909 4691' CD *** DB 'Illegal ORG Error', CR, 0
910 4692' 49 6C 8C 65 67
911 4693' 41 8C 24 4F 52
912 4694' 45 24 45 72 72
913 4695' 67 72 80 80
914 4696' 21 *** ext1: LD A, (pass)
915 4697' CD *** AND A
916 4698' CD *** CALL NZ, CLOSE
917 4699' CD *** JP, HOT
918 4700' 21 *** fErr: LD HL, C055
919 4701' CD *** CALL puts
920 4702' CD *** RET
921 4703' CD *** DB 'Illegal ORG Error', CR, 0
922 4704' 49 6C 8C 65 67
923 4705' 41 8C 24 4F 52
924 4706' 45 24 45 72 72
925 4707' 67 72 80 80
926 4708' 21 *** ext1: LD A, (pass)
927 4709' CD *** AND A
928 4710' CD *** CALL NZ, CLOSE
929 4711' CD *** JP, HOT
930 4712' 21 *** fErr: LD HL, C055
931 4713' CD *** CALL puts
932 4714' CD *** RET
933 4715' CD *** DB 'Illegal ORG Error', CR, 0
934 4716' 49 6C 8C 65 67
935 4717' 41 8C 24 4F 52
936 4718' 45 24 45 72 72
937 4719' 67 72 80 80
938 4720' 21 *** ext1: LD A, (pass)
939 4721' CD *** AND A
940 4722' CD *** CALL NZ, CLOSE
941 4723' CD *** JP, HOT
942 4724' 21 *** fErr: LD HL, C055
943 4725' CD *** CALL puts
944 4726' CD *** RET
945 4727' CD *** DB 'Illegal ORG Error', CR, 0
946 4728' 49 6C 8C 65 67
947 4729' 41 8C 24 4F 52
948 4730' 45 24 45 72 72
949 4731' 67 72 80 80
950 4732' 21 *** ext1: LD A, (pass)
951 4733' CD *** AND A
952 4734' CD *** CALL NZ, CLOSE
953 4735' CD *** JP, HOT
954 4736' 21 *** fErr: LD HL, C055
955 4737' CD *** CALL puts
956 4738' CD *** RET
957 4739' CD *** DB 'Illegal ORG Error', CR, 0
958 4740' 49 6C 8C 65 67
959 4741' 41 8C 24 4F 52
960 4742' 45 24 45 72 72
961 4743' 67 72 80 80
962 4744' 21 *** ext1: LD A, (pass)
963 4745' CD *** AND A
964 4746' CD *** CALL NZ, CLOSE
965 4747' CD *** JP, HOT
966 4748' 21 *** fErr: LD HL, C055
967 4749' CD *** CALL puts
968 4750' CD *** RET
969 4751' CD *** DB 'Illegal ORG Error', CR, 0
970 4752' 49 6C 8C 65 67
971 4753' 41 8C 24 4F 52
972 4754' 45 24 45 72 72
973 4755' 67 72 80 80
974 4756' 21 *** ext1: LD A, (pass)
975 4757' CD *** AND A
976 4758' CD *** CALL NZ, CLOSE
977 4759' CD *** JP, HOT
978 4760' 21 *** fErr: LD HL, C055
979 4761' CD *** CALL puts
980 4762' CD *** RET
981 4763' CD *** DB 'Illegal ORG Error', CR, 0
982 4764' 49 6C 8C 65 67
983 4765' 41 8C 24 4F 52
984 4766' 45 24 45 72 72
985 4767' 67 72 80 80
986 4768' 21 *** ext1: LD A, (pass)
987 4769' CD *** AND A
988 4770' CD *** CALL NZ, CLOSE
989 4771' CD *** JP, HOT
990 4772' 21 *** fErr: LD HL, C055
991 4773' CD *** CALL puts
992 4774' CD *** RET
993 4775' CD *** DB 'Illegal ORG Error', CR, 0
994 4776' 49 6C 8C 65 67
995 4777' 41 8C 24 4F 52
996 4778' 45 24 45 72 72
997 4779' 67 72 80 80
998 4780' 21 *** ext1: LD A, (pass)
999 4781' CD *** AND A
1000 4782' CD *** CALL NZ, CLOSE
1001 4783' CD *** JP, HOT

```

```

916 46AF' 06 14 : fcopy: LD B, LENFL
917 46B0' 06 14 : fcopy: LD B, LENFL
918 46B1' 06 14 : fcopy: LD B, LENFL
919 46B2' 06 14 : fcopy: LD B, LENFL
920 46B3' 06 14 : fcopy: LD B, LENFL
921 46B4' 06 14 : fcopy: LD B, LENFL
922 46B5' 06 14 : fcopy: LD B, LENFL
923 46B6' 06 14 : fcopy: LD B, LENFL
924 46B7' 06 14 : fcopy: LD B, LENFL
925 46B8' 06 14 : fcopy: LD B, LENFL
926 46B9' 06 14 : fcopy: LD B, LENFL
927 46BA' 06 14 : fcopy: LD B, LENFL
928 46BB' 06 14 : fcopy: LD B, LENFL
929 46BC' 06 14 : fcopy: LD B, LENFL
930 46BD' 06 14 : fcopy: LD B, LENFL
931 46BE' 06 14 : fcopy: LD B, LENFL
932 46BF' 06 14 : fcopy: LD B, LENFL
933 46C0' 06 14 : fcopy: LD B, LENFL
934 46C1' 06 14 : fcopy: LD B, LENFL
935 46C2' 06 14 : fcopy: LD B, LENFL
936 46C3' 06 14 : fcopy: LD B, LENFL
937 46C4' 06 14 : fcopy: LD B, LENFL
938 46C5' 06 14 : fcopy: LD B, LENFL
939 46C6' 06 14 : fcopy: LD B, LENFL
940 46C7' 06 14 : fcopy: LD B, LENFL
941 46C8' 06 14 : fcopy: LD B, LENFL
942 46C9' 06 14 : fcopy: LD B, LENFL
943 46CA' 06 14 : fcopy: LD B, LENFL
944 46CB' 06 14 : fcopy: LD B, LENFL
945 46CC' 06 14 : fcopy: LD B, LENFL
946 46CD' 06 14 : fcopy: LD B, LENFL
947 46CE' 06 14 : fcopy: LD B, LENFL
948 46CF' 06 14 : fcopy: LD B, LENFL
949 46D0' 06 14 : fcopy: LD B, LENFL
950 46D1' 06 14 : fcopy: LD B, LENFL
951 46D2' 06 14 : fcopy: LD B, LENFL
952 46D3' 06 14 : fcopy: LD B, LENFL
953 46D4' 06 14 : fcopy: LD B, LENFL
954 46D5' 06 14 : fcopy: LD B, LENFL
955 46D6' 06 14 : fcopy: LD B, LENFL
956 46D7' 06 14 : fcopy: LD B, LENFL
957 46D8' 06 14 : fcopy: LD B, LENFL
958 46D9' 06 14 : fcopy: LD B, LENFL
959 46DA' 06 14 : fcopy: LD B, LENFL
960 46DB' 06 14 : fcopy: LD B, LENFL
961 46DC' 06 14 : fcopy: LD B, LENFL
962 46DD' 06 14 : fcopy: LD B, LENFL
963 46DE' 06 14 : fcopy: LD B, LENFL
964 46DF' 06 14 : fcopy: LD B, LENFL
965 46E0' 06 14 : fcopy: LD B, LENFL
966 46E1' 06 14 : fcopy: LD B, LENFL
967 46E2' 06 14 : fcopy: LD B, LENFL
968 46E3' 06 14 : fcopy: LD B, LENFL
969 46E4' 06 14 : fcopy: LD B, LENFL
970 46E5' 06 14 : fcopy: LD B, LENFL
971 46E6' 06 14 : fcopy: LD B, LENFL
972 46E7' 06 14 : fcopy: LD B, LENFL
973 46E8' 06 14 : fcopy: LD B, LENFL
974 46E9' 06 14 : fcopy: LD B, LENFL
975 46EA' 06 14 : fcopy: LD B, LENFL
976 46EB' 06 14 : fcopy: LD B, LENFL
977 46EC' 06 14 : fcopy: LD B, LENFL
978 46ED' 06 14 : fcopy: LD B, LENFL
979 46EE' 06 14 : fcopy: LD B, LENFL
980 46EF' 06 14 : fcopy: LD B, LENFL
981 46F0' 06 14 : fcopy: LD B, LENFL
982 46F1' 06 14 : fcopy: LD B, LENFL
983 46F2' 06 14 : fcopy: LD B, LENFL
984 46F3' 06 14 : fcopy: LD B, LENFL
985 46F4' 06 14 : fcopy: LD B, LENFL
986 46F5' 06 14 : fcopy: LD B, LENFL
987 46F6' 06 14 : fcopy: LD B, LENFL
988 46F7' 06 14 : fcopy: LD B, LENFL
989 46F8' 06 14 : fcopy: LD B, LENFL
990 46F9' 06 14 : fcopy: LD B, LENFL
991 46FA' 06 14 : fcopy: LD B, LENFL
992 46FB' 06 14 : fcopy: LD B, LENFL
993 46FC' 06 14 : fcopy: LD B, LENFL
994 46FD' 06 14 : fcopy: LD B, LENFL
995 46FE' 06 14 : fcopy: LD B, LENFL
996 46FF' 06 14 : fcopy: LD B, LENFL
997 4700' 06 14 : fcopy: LD B, LENFL
998 4701' 06 14 : fcopy: LD B, LENFL
999 4702' 06 14 : fcopy: LD B, LENFL
1000 4703' 06 14 : fcopy: LD B, LENFL

```

```

988 4704' 06 14 : fcopy: LD B, LENFL
989 4705' 06 14 : fcopy: LD B, LENFL
990 4706' 06 14 : fcopy: LD B, LENFL
991 4707' 06 14 : fcopy: LD B, LENFL
992 4708' 06 14 : fcopy: LD B, LENFL
993 4709' 06 14 : fcopy: LD B, LENFL
994 4710' 06 14 : fcopy: LD B, LENFL
995 4711' 06 14 : fcopy: LD B, LENFL
996 4712' 06 14 : fcopy: LD B, LENFL
997 4713' 06 14 : fcopy: LD B, LENFL
998 4714' 06 14 : fcopy: LD B, LENFL
999 4715' 06 14 : fcopy: LD B, LENFL
1000 4716' 06 14 : fcopy: LD B, LENFL

```

リスト3

```

1 0000' : Linker For W-ZEDA
2 0001' :
3 0002' :
4 0003' :
5 0004' :
6 0005' :
7 0006' :
8 0007' :
9 0008' :
10 0009' :
11 0010' :
12 0011' :
13 0012' :
14 0013' :
15 0014' :
16 0015' :
17 0016' :
18 0017' :
19 0018' :
20 0019' :
21 0020' :
22 0021' :
23 0022' :
24 0023' :
25 0024' :
26 0025' :
27 0026' :
28 0027' :
29 0028' :
30 0029' :
31 0030' :
32 0031' :
33 0032' :
34 0033' :
35 0034' :
36 0035' :
37 0036' :
38 0037' :
39 0038' :
40 0039' :
41 0040' :
42 0041' :
43 0042' :
44 0043' :
45 0044' :
46 0045' :
47 0046' :
48 0047' :
49 0048' :
50 0049' :
51 0050' :
52 0051' :
53 0052' :
54 0053' :
55 0054' :
56 0055' :
57 0056' :
58 0057' :
59 0058' :
60 0059' :
61 0060' :
62 0061' :
63 0062' :
64 0063' :
65 0064' :
66 0065' :
67 0066' :
68 0067' :
69 0068' :
70 0069' :
71 0070' :
72 0071' :
73 0072' :
74 0073' :
75 0074' :
76 0075' :
77 0076' :
78 0077' :
79 0078' :
80 0079' :
81 0080' :
82 0081' :
83 0082' :
84 0083' :
85 0084' :
86 0085' :
87 0086' :
88 0087' :
89 0088' :
90 0089' :
91 0090' :
92 0091' :
93 0092' :
94 0093' :
95 0094' :
96 0095' :
97 0096' :
98 0097' :
99 0098' :
100 0099' :

```

```

84 0000' : Linker For W-ZEDA
85 0001' :
86 0002' :
87 0003' :
88 0004' :
89 0005' :
90 0006' :
91 0007' :
92 0008' :
93 0009' :
94 0010' :
95 0011' :
96 0012' :
97 0013' :
98 0014' :
99 0015' :
100 0016' :
101 0017' :
102 0018' :
103 0019' :
104 0020' :
105 0021' :
106 0022' :
107 0023' :
108 0024' :
109 0025' :
110 0026' :
111 0027' :
112 0028' :
113 0029' :
114 0030' :
115 0031' :
116 0032' :
117 0033' :
118 0034' :
119 0035' :
120 0036' :
121 0037' :
122 0038' :
123 0039' :
124 0040' :
125 0041' :
126 0042' :
127 0043' :
128 0044' :
129 0045' :
130 0046' :
131 0047' :
132 0048' :
133 0049' :
134 0050' :
135 0051' :
136 0052' :
137 0053' :
138 0054' :
139 0055' :
140 0056' :
141 0057' :
142 0058' :
143 0059' :
144 0060' :
145 0061' :
146 0062' :
147 0063' :
148 0064' :
149 0065' :
150 0066' :
151 0067' :
152 0068' :
153 0069' :
154 0070' :
155 0071' :
156 0072' :
157 0073' :
158 0074' :
159 0075' :
160 0076' :
161 0077' :
162 0078' :
163 0079' :
164 0080' :
165 0081' :

```

```

166 0082' :
167 0083' :
168 0084' :
169 0085' :
170 0086' :
171 0087' :
172 0088' :
173 0089' :
174 0090' :
175 0091' :
176 0092' :
177 0093' :
178 0094' :
179 0095' :
180 0096' :
181 0097' :
182 0098' :
183 0099' :
184 0100' :
185 0101' :
186 0102' :
187 0103' :
188 0104' :
189 0105' :
190 0106' :
191 0107' :
192 0108' :
193 0109' :
194 0110' :
195 0111' :
196 0112' :
197 0113' :
198 0114' :
199 0115' :
200 0116' :
201 0117' :
202 0118' :
203 0119' :
204 0120' :
205 0121' :
206 0122' :
207 0123' :
208 0124' :
209 0125' :
210 0126' :
211 0127' :
212 0128' :
213 0129' :
214 0130' :
215 0131' :
216 0132' :
217 0133' :
218 0134' :
219 0135' :
220 0136' :
221 0137' :
222 0138' :
223 0139' :
224 0140' :
225 0141' :
226 0142' :
227 0143' :
228 0144' :
229 0145' :
230 0146' :
231 0147' :
232 0148' :
233 0149' :
234 0150' :
235 0151' :
236 0152' :
237 0153' :
238 0154' :
239 0155' :
240 0156' :
241 0157' :
242 0158' :
243 0159' :
244 0160' :
245 0161' :
246 0162' :
247 0163' :
248 0164' :

```



```

737 044B' 78      LD A,B
738 044C' 81      OR C
739 044D' 20 F7   JR NZ,DEF1
740 044E'          ; Chain last struct with New struct
741 044F'          DFC HL
742 0450' 28      DEC HL
743 0451' 28      DEC HL
744 0452' 20 48 ** LD HL,(L,LBLPTR)
745 0453' CD ***  CALL POKE_BC
746 0454'          ; ** Make Struct of Labels **
747 0455'          LD HL,(LBLPTR)
748 0456' 2A ***  LD BC,0
749 0457' 01 00 ** CALL POKE_BC
750 0458' CD ***  CALL POKE_I
751 0459' 2A ***  LD A,(SV_FIG)
752 0460' CD ***  LD BC,(SV_VAR)
753 0461' 20 48 ** CALL POKE_BC
754 0462'          ; ** Make Struct of Labels **
755 0463'          LD HL,(LBLPTR)
756 0464' 2A ***  LD BC,0
757 0465' 01 00 ** CALL POKE_BC
758 0466' CD ***  CALL POKE_I
759 0467' 2A ***  LD A,(SV_FIG)
760 0468' CD ***  LD BC,(SV_VAR)
761 0469'          ; ** Make Struct of Labels **
762 0470'          LD HL,(LBLPTR)
763 0471' 2A ***  LD BC,0
764 0472' 01 00 ** CALL POKE_BC
765 0473' CD ***  CALL POKE_I
766 0474' 2A ***  LD A,(SV_FIG)
767 0475' CD ***  LD BC,(SV_VAR)
768 0476'          ; ** Make Struct of Labels **
769 0477'          LD HL,(LBLPTR)
770 0478' 2A ***  LD BC,0
771 0479' 01 00 ** CALL POKE_BC
772 0480' CD ***  CALL POKE_I
773 0481' 2A ***  LD A,(SV_FIG)
774 0482' CD ***  LD BC,(SV_VAR)
775 0483'          ; ** Make Struct of Labels **
776 0484'          LD HL,(LBLPTR)
777 0485' 2A ***  LD BC,0
778 0486' 01 00 ** CALL POKE_BC
779 0487' CD ***  CALL POKE_I
780 0488' 2A ***  LD A,(SV_FIG)
781 0489' CD ***  LD BC,(SV_VAR)
782 0490'          ; ** Make Struct of Labels **
783 0491'          LD HL,(LBLPTR)
784 0492' 2A ***  LD BC,0
785 0493' 01 00 ** CALL POKE_BC
786 0494' CD ***  CALL POKE_I
787 0495' 2A ***  LD A,(SV_FIG)
788 0496' CD ***  LD BC,(SV_VAR)
789 0497'          ; ** Make Struct of Labels **
790 0498'          LD HL,(LBLPTR)
791 0499' 2A ***  LD BC,0
792 0500' 01 00 ** CALL POKE_BC
793 0501' CD ***  CALL POKE_I
794 0502' 2A ***  LD A,(SV_FIG)
795 0503' CD ***  LD BC,(SV_VAR)
796 0504'          ; ** Make Struct of Labels **
797 0505'          LD HL,(LBLPTR)
798 0506' 2A ***  LD BC,0
799 0507' 01 00 ** CALL POKE_BC
800 0508' CD ***  CALL POKE_I
801 0509' 2A ***  LD A,(SV_FIG)
802 0510' CD ***  LD BC,(SV_VAR)
803 0511'          ; ** Make Struct of Labels **
804 0512'          LD HL,(LBLPTR)
805 0513' 2A ***  LD BC,0
806 0514' 01 00 ** CALL POKE_BC
807 0515' CD ***  CALL POKE_I
808 0516' 2A ***  LD A,(SV_FIG)
809 0517' CD ***  LD BC,(SV_VAR)
810 0518'          ; ** Make Struct of Labels **
811 0519'          LD HL,(LBLPTR)
812 0520' 2A ***  LD BC,0
813 0521' 01 00 ** CALL POKE_BC
814 0522' CD ***  CALL POKE_I
815 0523' 2A ***  LD A,(SV_FIG)
816 0524' CD ***  LD BC,(SV_VAR)
817 0525'          ; ** Make Struct of Labels **
818 0526'          LD HL,(LBLPTR)
819 0527' 2A ***  LD BC,0
820 0528' 01 00 ** CALL POKE_BC
821 0529' CD ***  CALL POKE_I
822 0530' 2A ***  LD A,(SV_FIG)
823 0531' CD ***  LD BC,(SV_VAR)
824 0532'          ; ** Make Struct of Labels **
825 0533'          LD HL,(LBLPTR)
826 0534' 2A ***  LD BC,0
827 0535' 01 00 ** CALL POKE_BC
828 0536' CD ***  CALL POKE_I
829 0537' 2A ***  LD A,(SV_FIG)
830 0538' CD ***  LD BC,(SV_VAR)
831 0539'          ; ** Make Struct of Labels **
832 0540'          LD HL,(LBLPTR)
833 0541' 2A ***  LD BC,0
834 0542' 01 00 ** CALL POKE_BC
835 0543' CD ***  CALL POKE_I
836 0544' 2A ***  LD A,(SV_FIG)
837 0545' CD ***  LD BC,(SV_VAR)
838 0546'          ; ** Make Struct of Labels **
839 0547'          LD HL,(LBLPTR)
840 0548' 2A ***  LD BC,0
841 0549' 01 00 ** CALL POKE_BC
842 0550' CD ***  CALL POKE_I
843 0551' 2A ***  LD A,(SV_FIG)
844 0552' CD ***  LD BC,(SV_VAR)
845 0553'          ; ** Make Struct of Labels **
846 0554'          LD HL,(LBLPTR)
847 0555' 2A ***  LD BC,0
848 0556' 01 00 ** CALL POKE_BC
849 0557' CD ***  CALL POKE_I
850 0558' 2A ***  LD A,(SV_FIG)
851 0559' CD ***  LD BC,(SV_VAR)
852 0560'          ; ** Make Struct of Labels **
853 0561'          LD HL,(LBLPTR)
854 0562' 2A ***  LD BC,0
855 0563' 01 00 ** CALL POKE_BC
856 0564' CD ***  CALL POKE_I
857 0565' 2A ***  LD A,(SV_FIG)
858 0566' CD ***  LD BC,(SV_VAR)
859 0567'          ; ** Make Struct of Labels **
860 0568'          LD HL,(LBLPTR)
861 0569' 2A ***  LD BC,0
862 0570' 01 00 ** CALL POKE_BC
863 0571' CD ***  CALL POKE_I
864 0572' 2A ***  LD A,(SV_FIG)
865 0573' CD ***  LD BC,(SV_VAR)
866 0574'          ; ** Make Struct of Labels **
867 0575'          LD HL,(LBLPTR)
868 0576' 2A ***  LD BC,0
869 0577' 01 00 ** CALL POKE_BC
870 0578' CD ***  CALL POKE_I
871 0579' 2A ***  LD A,(SV_FIG)
872 0580' CD ***  LD BC,(SV_VAR)
873 0581'          ; ** Make Struct of Labels **
874 0582'          LD HL,(LBLPTR)
875 0583' 2A ***  LD BC,0
876 0584' 01 00 ** CALL POKE_BC
877 0585' CD ***  CALL POKE_I
878 0586' 2A ***  LD A,(SV_FIG)
879 0587' CD ***  LD BC,(SV_VAR)
880 0588'          ; ** Make Struct of Labels **
881 0589'          LD HL,(LBLPTR)
882 0590' 2A ***  LD BC,0
883 0591' 01 00 ** CALL POKE_BC
884 0592' CD ***  CALL POKE_I
885 0593' 2A ***  LD A,(SV_FIG)
886 0594' CD ***  LD BC,(SV_VAR)
887 0595'          ; ** Make Struct of Labels **
888 0596'          LD HL,(LBLPTR)
889 0597' 2A ***  LD BC,0
890 0598' 01 00 ** CALL POKE_BC
891 0599' CD ***  CALL POKE_I
892 0600' 2A ***  LD A,(SV_FIG)
893 0601' CD ***  LD BC,(SV_VAR)
894 0602'          ; ** Make Struct of Labels **
895 0603'          LD HL,(LBLPTR)
896 0604' 2A ***  LD BC,0
897 0605' 01 00 ** CALL POKE_BC
898 0606' CD ***  CALL POKE_I
899 0607' 2A ***  LD A,(SV_FIG)
900 0608' CD ***  LD BC,(SV_VAR)
901 0609'          ; ** Make Struct of Labels **
902 0610'          LD HL,(LBLPTR)
903 0611' 2A ***  LD BC,0
904 0612' 01 00 ** CALL POKE_BC
905 0613' CD ***  CALL POKE_I
906 0614' 2A ***  LD A,(SV_FIG)
907 0615' CD ***  LD BC,(SV_VAR)
908 0616'          ; ** Make Struct of Labels **
909 0617'          LD HL,(LBLPTR)
910 0618' 2A ***  LD BC,0
911 0619' 01 00 ** CALL POKE_BC
912 0620' CD ***  CALL POKE_I
913 0621' 2A ***  LD A,(SV_FIG)
914 0622' CD ***  LD BC,(SV_VAR)
915 0623'          ; ** Make Struct of Labels **
916 0624'          LD HL,(LBLPTR)
917 0625' 2A ***  LD BC,0
918 0626' 01 00 ** CALL POKE_BC
919 0627' CD ***  CALL POKE_I
920 0628' 2A ***  LD A,(SV_FIG)
921 0629' CD ***  LD BC,(SV_VAR)
922 0630'          ; ** Make Struct of Labels **
923 0631'          LD HL,(LBLPTR)
924 0632' 2A ***  LD BC,0
925 0633' 01 00 ** CALL POKE_BC
926 0634' CD ***  CALL POKE_I
927 0635' 2A ***  LD A,(SV_FIG)
928 0636' CD ***  LD BC,(SV_VAR)
929 0637'          ; ** Make Struct of Labels **
930 0638'          LD HL,(LBLPTR)
931 0639' 2A ***  LD BC,0
932 0640' 01 00 ** CALL POKE_BC
933 0641' CD ***  CALL POKE_I
934 0642' 2A ***  LD A,(SV_FIG)
935 0643' CD ***  LD BC,(SV_VAR)
936 0644'          ; ** Make Struct of Labels **
937 0645'          LD HL,(LBLPTR)
938 0646' 2A ***  LD BC,0
939 0647' 01 00 ** CALL POKE_BC
940 0648' CD ***  CALL POKE_I
941 0649' 2A ***  LD A,(SV_FIG)
942 0650' CD ***  LD BC,(SV_VAR)
943 0651'          ; ** Make Struct of Labels **
944 0652'          LD HL,(LBLPTR)
945 0653' 2A ***  LD BC,0
946 0654' 01 00 ** CALL POKE_BC
947 0655' CD ***  CALL POKE_I
948 0656' 2A ***  LD A,(SV_FIG)
949 0657' CD ***  LD BC,(SV_VAR)
950 0658'          ; ** Make Struct of Labels **
951 0659'          LD HL,(LBLPTR)
952 0660' 2A ***  LD BC,0
953 0661' 01 00 ** CALL POKE_BC
954 0662' CD ***  CALL POKE_I
955 0663' 2A ***  LD A,(SV_FIG)
956 0664' CD ***  LD BC,(SV_VAR)
957 0665'          ; ** Make Struct of Labels **
958 0666'          LD HL,(LBLPTR)
959 0667' 2A ***  LD BC,0
960 0668' 01 00 ** CALL POKE_BC
961 0669' CD ***  CALL POKE_I
962 0670' 2A ***  LD A,(SV_FIG)
963 0671' CD ***  LD BC,(SV_VAR)
964 0672'          ; ** Make Struct of Labels **
965 0673'          LD HL,(LBLPTR)
966 0674' 2A ***  LD BC,0
967 0675' 01 00 ** CALL POKE_BC
968 0676' CD ***  CALL POKE_I
969 0677' 2A ***  LD A,(SV_FIG)
970 0678' CD ***  LD BC,(SV_VAR)
971 0679'          ; ** Make Struct of Labels **
972 0680'          LD HL,(LBLPTR)
973 0681' 2A ***  LD BC,0
974 0682' 01 00 ** CALL POKE_BC
975 0683' CD ***  CALL POKE_I
976 0684' 2A ***  LD A,(SV_FIG)
977 0685' CD ***  LD BC,(SV_VAR)
978 0686'          ; ** Make Struct of Labels **
979 0687'          LD HL,(LBLPTR)
980 0688' 2A ***  LD BC,0
981 0689' 01 00 ** CALL POKE_BC
982 0690' CD ***  CALL POKE_I
983 0691' 2A ***  LD A,(SV_FIG)
984 0692' CD ***  LD BC,(SV_VAR)
985 0693'          ; ** Make Struct of Labels **
986 0694'          LD HL,(LBLPTR)
987 0695' 2A ***  LD BC,0
988 0696' 01 00 ** CALL POKE_BC
989 0697' CD ***  CALL POKE_I
990 0698' 2A ***  LD A,(SV_FIG)
991 0699' CD ***  LD BC,(SV_VAR)
992 0700'          ; ** Make Struct of Labels **
993 0701'          LD HL,(LBLPTR)
994 0702' 2A ***  LD BC,0
995 0703' 01 00 ** CALL POKE_BC
996 0704' CD ***  CALL POKE_I
997 0705' 2A ***  LD A,(SV_FIG)
998 0706' CD ***  LD BC,(SV_VAR)
999 0707'          ; ** Make Struct of Labels **
1000 0708'          LD HL,(LBLPTR)
1001 0709' 2A ***  LD BC,0
1002 0710' 01 00 ** CALL POKE_BC
1003 0711' CD ***  CALL POKE_I
1004 0712' 2A ***  LD A,(SV_FIG)
1005 0713' CD ***  LD BC,(SV_VAR)
1006 0714'          ; ** Make Struct of Labels **
1007 0715'          LD HL,(LBLPTR)
1008 0716' 2A ***  LD BC,0
1009 0717' 01 00 ** CALL POKE_BC
1010 0718' CD ***  CALL POKE_I
1011 0719' 2A ***  LD A,(SV_FIG)
1012 0720' CD ***  LD BC,(SV_VAR)
1013 0721'          ; ** Make Struct of Labels **
1014 0722'          LD HL,(LBLPTR)
1015 0723' 2A ***  LD BC,0
1016 0724' 01 00 ** CALL POKE_BC
1017 0725' CD ***  CALL POKE_I
1018 0726' 2A ***  LD A,(SV_FIG)
1019 0727' CD ***  LD BC,(SV_VAR)
1020 0728'          ; ** Make Struct of Labels **
1021 0729'          LD HL,(LBLPTR)
1022 0730' 2A ***  LD BC,0
1023 0731' 01 00 ** CALL POKE_BC
1024 0732' CD ***  CALL POKE_I
1025 0733' 2A ***  LD A,(SV_FIG)
1026 0734' CD ***  LD BC,(SV_VAR)
1027 0735'          ; ** Make Struct of Labels **
1028 0736'          LD HL,(LBLPTR)
1029 0737' 2A ***  LD BC,0
1030 0738' 01 00 ** CALL POKE_BC
1031 0739' CD ***  CALL POKE_I
1032 0740' 2A ***  LD A,(SV_FIG)
1033 0741' CD ***  LD BC,(SV_VAR)
1034 0742'          ; ** Make Struct of Labels **
1035 0743'          LD HL,(LBLPTR)
1036 0744' 2A ***  LD BC,0
1037 0745' 01 00 ** CALL POKE_BC
1038 0746' CD ***  CALL POKE_I
1039 0747' 2A ***  LD A,(SV_FIG)
1040 0748' CD ***  LD BC,(SV_VAR)
1041 0749'          ; ** Make Struct of Labels **
1042 0750'          LD HL,(LBLPTR)
1043 0751' 2A ***  LD BC,0
1044 0752' 01 00 ** CALL POKE_BC
1045 0753' CD ***  CALL POKE_I
1046 0754' 2A ***  LD A,(SV_FIG)
1047 0755' CD ***  LD BC,(SV_VAR)
1048 0756'          ; ** Make Struct of Labels **
1049 0757'          LD HL,(LBLPTR)
1050 0758' 2A ***  LD BC,0
1051 0759' 01 00 ** CALL POKE_BC
1052 0760' CD ***  CALL POKE_I
1053 0761' 2A ***  LD A,(SV_FIG)
1054 0762' CD ***  LD BC,(SV_VAR)
1055 0763'          ; ** Make Struct of Labels **
1056 0764'          LD HL,(LBLPTR)
1057 0765' 2A ***  LD BC,0
1058 0766' 01 00 ** CALL POKE_BC
1059 0767' CD ***  CALL POKE_I
1060 0768' 2A ***  LD A,(SV_FIG)
1061 0769' CD ***  LD BC,(SV_VAR)
1062 0770'          ; ** Make Struct of Labels **
1063 0771'          LD HL,(LBLPTR)
1064 0772' 2A ***  LD BC,0
1065 0773' 01 00 ** CALL POKE_BC
1066 0774' CD ***  CALL POKE_I
1067 0775' 2A ***  LD A,(SV_FIG)
1068 0776' CD ***  LD BC,(SV_VAR)
1069 0777'          ; ** Make Struct of Labels **
1070 0778'          LD HL,(LBLPTR)
1071 0779' 2A ***  LD BC,0
1072 0780' 01 00 ** CALL POKE_BC
1073 0781' CD ***  CALL POKE_I
1074 0782' 2A ***  LD A,(SV_FIG)
1075 0783' CD ***  LD BC,(SV_VAR)
1076 0784'          ; ** Make Struct of Labels **
1077 0785'          LD HL,(LBLPTR)
1078 0786' 2A ***  LD BC,0
1079 0787' 01 00 ** CALL POKE_BC
1080 0788' CD ***  CALL POKE_I
1081 0789' 2A ***  LD A,(SV_FIG)
1082 0790' CD ***  LD BC,(SV_VAR)
1083 0791'          ; ** Make Struct of Labels **
1084 0792'          LD HL,(LBLPTR)
1085 0793' 2A ***  LD BC,0
1086 0794' 01 00 ** CALL POKE_BC
1087 0795' CD ***  CALL POKE_I
1088 0796' 2A ***  LD A,(SV_FIG)
1089 0797' CD ***  LD BC,(SV_VAR)
1090 0798'          ; ** Make Struct of Labels **
1091 0799'          LD HL,(LBLPTR)
1092 0800' 2A ***  LD BC,0
1093 0801' 01 00 ** CALL POKE_BC
1094 0802' CD ***  CALL POKE_I
1095 0803' 2A ***  LD A,(SV_FIG)
1096 0804' CD ***  LD BC,(SV_VAR)
1097 0805'          ; ** Make Struct of Labels **
1098 0806'          LD HL,(LBLPTR)
1099 0807' 2A ***  LD BC,0
1100 0808' 01 00 ** CALL POKE_BC
1101 0809' CD ***  CALL POKE_I
1102 0810' 2A ***  LD A,(SV_FIG)
1103 0811' CD ***  LD BC,(SV_VAR)
1104 0812'          ; ** Make Struct of Labels **
1105 0813'          LD HL,(LBLPTR)
1106 0814' 2A ***  LD BC,0
1107 0815' 01 00 ** CALL POKE_BC
1108 0816' CD ***  CALL POKE_I
1109 0817' 2A ***  LD A,(SV_FIG)
1110 0818' CD ***  LD BC,(SV_VAR)
1111 0819'          ; ** Make Struct of Labels **
1112 0820'          LD HL,(LBLPTR)
1113 0821' 2A ***  LD BC,0
1114 0822' 01 00 ** CALL POKE_BC
1115 0823' CD ***  CALL POKE_I
1116 0824' 2A ***  LD A,(SV_FIG)
1117 0825' CD ***  LD BC,(SV_VAR)
1118 0826'          ; ** Make Struct of Labels **
1119 0827'          LD HL,(LBLPTR)
1120 0828' 2A ***  LD BC,0
1121 0829' 01 00 ** CALL POKE_BC
1122 0830' CD ***  CALL POKE_I
1123 0831' 2A ***  LD A,(SV_FIG)
1124 0832' CD ***  LD BC,(SV_VAR)
1125 0833'          ; ** Make Struct of Labels **
1126 0834'          LD HL,(LBLPTR)
1127 0835' 2A ***  LD BC,0
1128 0836' 01 00 ** CALL POKE_BC
1129 0837' CD ***  CALL POKE_I
1130 0838' 2A ***  LD A,(SV_FIG)
1131 0839' CD ***  LD BC,(SV_VAR)
1132 0840'          ; ** Make Struct of Labels **
1133 0841'          LD HL,(LBLPTR)
1134 0842' 2A ***  LD BC,0
1135 0843' 01 00 ** CALL POKE_BC
1136 0844' CD ***  CALL POKE_I
1137 0845' 2A ***  LD A,(SV_FIG)
1138 0846' CD ***  LD BC,(SV_VAR)
1139 0847'          ; ** Make Struct of Labels **
1140 0848'          LD HL,(LBLPTR)
1141 0849' 2A ***  LD BC,0
1142 0850' 01 00 ** CALL POKE_BC
1143 0851' CD ***  CALL POKE_I
1144 0852' 2A ***  LD A,(SV_FIG)
1145 0853' CD ***  LD BC,(SV_VAR)
1146 0854'          ; ** Make Struct of Labels **
1147 0855'          LD HL,(LBLPTR)
1148 0856' 2A ***  LD BC,0
1149 0857' 01 00 ** CALL POKE_BC
1150 0858' CD ***  CALL POKE_I
1151 0859' 2A ***  LD A,(SV_FIG)
1152 0860' CD ***  LD BC,(SV_VAR)
1153 0861'          ; ** Make Struct of Labels **
1154 0862'          LD HL,(LBLPTR)
1155 0863' 2A ***  LD BC,0
1156 0864' 01 00 ** CALL POKE_BC
1157 0865' CD ***  CALL POKE_I
1158 0866' 2A ***  LD A,(SV_FIG)
1159 0867' CD ***  LD BC,(SV_VAR)
1160 0868'          ; ** Make Struct of Labels **
1161 0869'          LD HL,(LBLPTR)
1162 0870' 2A ***  LD BC,0
1163 0871' 01 00 ** CALL POKE_BC
1164 0872' CD ***  CALL POKE_I
1165 0873' 2A ***  LD A,(SV_FIG)
1166 0874' CD ***  LD BC,(SV_VAR)
1167 0875'          ; ** Make Struct of Labels **
1168 0876'          LD HL,(LBLPTR)
1169 0877' 2A ***  LD BC,0
1170 0878' 01 00 ** CALL POKE_BC
1171 0879' CD ***  CALL POKE_I
1172 0880' 2A ***  LD A,(SV_FIG)
1173 0881' CD ***  LD BC,(SV_VAR)
1174 0882'          ; ** Make Struct of Labels **
1175 0883'          LD HL,(LBLPTR)
1176 0884' 2A ***  LD BC,0
1177 0885' 01 00 ** CALL POKE_BC
1178 0886' CD ***  CALL POKE_I
1179 0887' 2A ***  LD A,(SV_FIG)
1180 0888' CD ***  LD BC,(SV_VAR)
1181 0889'          ; ** Make Struct of Labels **
1182 0890'          LD HL,(LBLPTR)
1183 0891' 2A ***  LD BC,0
1184 0892' 01 00 ** CALL POKE_BC
1185 0893' CD ***  CALL POKE_I
1186 0894' 2A ***  LD A,(SV_FIG)
1187 0895' CD ***  LD BC,(SV_VAR)
1188 0896'          ; ** Make Struct of Labels **
1189 0897'          LD HL,(LBLPTR)
1190 0898' 2A ***  LD BC,0
1191 0899' 01 00 ** CALL POKE_BC
1192 0900' CD ***  CALL POKE_I
1193 0901' 2A ***  LD A,(SV_FIG)
1194 0902' CD ***  LD BC,(SV_VAR)
1195 0903'          ; ** Make Struct of Labels **
1196 0904'          LD HL,(LBLPTR)
1197 0905' 2A ***  LD BC,0
1198 0906' 01 00 ** CALL POKE_BC
1199 0907' CD ***  CALL POKE_I
1200 0908' 2A ***  LD A,(SV_FIG)
1201 0909' CD ***  LD BC,(SV_VAR)
1202 0910'          ; ** Make Struct of Labels **
1203 0911'          LD HL,(LBLPTR)
1204 0912' 2A ***  LD BC,0
1205 0913' 01 00 ** CALL POKE_BC
1206 0914' CD ***  CALL POKE_I
1207 0915' 2A ***  LD A,(SV_FIG)
1208 0916' CD ***  LD BC,(SV_VAR)
1209 0917'          ; ** Make Struct of Labels **
1210 0918'          LD HL,(LBLPTR)
1211 0919' 2A ***  LD BC,0
1212 0920' 01 00 ** CALL POKE_BC
1213 0921' CD ***  CALL POKE_I
1214 0922' 2A ***  LD A,(SV_FIG)
1215 0923' CD ***  LD BC,(SV_VAR)
1216 0924'          ; ** Make Struct of Labels **
1217 0925'          LD HL,(LBLPTR)
1218 0926' 2A ***  LD BC,0
1219 0927' 01 00 ** CALL POKE_BC
1220 0928' CD ***  CALL POKE_I
1221 0929' 2A ***  LD A,(SV_FIG)
1222 0930' CD ***  LD BC,(SV_VAR)
1223 0931'          ; ** Make Struct of Labels **
1224 0932'          LD HL,(LBLPTR)
1225 0933' 2A ***  LD BC,0
1226 0934' 01 00 ** CALL POKE_BC
1227 0935' CD ***  CALL POKE_I
1228 0936' 2A ***  LD A,(SV_FIG)
1229 0937' CD ***  LD BC,(SV_VAR)
1230 0938'          ; ** Make Struct of Labels **
1231 0939'          LD HL,(LBLPTR)
1232 0940' 2A ***  LD BC,0
1233 0941' 01 00 ** CALL POKE_BC
1234 0942' CD ***  CALL POKE_I
1235 0943' 2A ***  LD A,(SV_FIG)
1236 0944' CD ***  LD BC,(SV_VAR)
1237 0945'          ; ** Make Struct of Labels **
1238 0946'          LD HL,(LBLPTR)
1239 0947' 2A ***  LD BC,0
1240 0948' 01 00 ** CALL POKE_BC
1241 0949' CD ***  CALL POKE_I
1242 0950' 2A ***  LD A,(SV_FIG)
1243 0951' CD ***  LD BC,(SV_VAR)
1244 0952'          ; ** Make Struct of Labels **
1245 0953'          LD HL,(LBLPTR)
1246 0954' 2A ***  LD BC,0
1247 0955' 01 00 ** CALL POKE_BC
1248 0956' CD ***  CALL POKE_I
1249 0957' 2A ***  LD A,(SV_FIG)
1250 0958' CD ***  LD BC,(SV_VAR)
1251 0959'          ; ** Make Struct of Labels **
1252 0960'          LD HL,(LBLPTR)
1253 0961' 2A ***  LD BC,0
1254 0962' 01 00 ** CALL POKE_BC
1255 0963' CD ***  CALL POKE_I
1256 0964' 2A ***  LD A,(SV_FIG)
1257 0965' CD ***  LD BC,(SV_VAR)
1258 0966'          ; ** Make Struct of Labels **
1259 0967'          LD HL,(LBLPTR)
1260 0968' 2A ***  LD BC,0
1261 0969' 01 00 ** CALL POKE_BC
1262 0970' CD ***  CALL POKE_I
1263 0971' 2A ***  LD A,(SV_FIG)
1264 0972' CD ***  LD BC,(SV_VAR)
1265 0973'          ; ** Make Struct of Labels **
1266 0974'          LD HL,(LBLPTR)
1267 0975' 2A ***  LD BC,0
1268 0976' 01 00 ** CALL POKE_BC
1269 0977' CD ***  CALL POKE_I
1270 0978' 2A ***  LD A,(SV_FIG)
1271 0979' CD ***  LD BC,(SV_VAR)
1272 0980'          ; ** Make Struct of Labels **
1273 0981'          LD HL,(LBLPTR)
1274 0982' 2A ***  LD BC,0
1275 0983' 01 00 ** CALL POKE_BC
1276 0984' CD ***  CALL POKE_I
1277 0985' 2A ***  LD A,(SV_FIG)
1278 0986' CD ***  LD BC,(SV_VAR)
1279 0987'          ; ** Make Struct of Labels **
1280 0988'          LD HL,(LBLPTR)
1281 0989' 2A ***  LD BC,0
1282 0990' 01 00 ** CALL POKE_BC
1283 0991' CD ***  CALL POKE_I
1284 0992' 2A ***  LD A,(SV_FIG)
1285 0993' CD ***  LD BC,(SV_VAR)
1286 0994'          ; ** Make Struct of Labels **
1287 0995'          LD HL,(LBLPTR)
1288 0996' 2A ***  LD BC,0
1289 0997' 01 00 ** CALL POKE_BC
1290 0998' CD ***  CALL POKE_I
1291 0999' 2A ***  LD A,(SV_FIG)
1300 1000' CD ***  LD BC,(SV_VAR)

```

```

1222 0785' C8          RET     NZ
1223 0786' 23 ***     LD      HL,(PTRC)
1224 0787' 3A ***     LD      A,(SEGND)
1225 0788' FE 02      CP      2
1226 078C' C8          RET
1227 078D' 2A ***     LD      HL,(PTRDT)
1228 0790' FE 03      CP      3
1229 0792' C8          RET
1230 0793' 2A ***     LD      HL,(PTRWK)
1231 0796' C0          RET
1232 0797'
1233 0797'
1234 0797'
1235 0797'
1236 0797' F5        INCPCH PUSH AF
1237 0798' E5        PUSH     HL
1238 0799' 3A ***     LD      A,(SEGND)
1239 079C' FE 03      CP      3
1240 079D' 2A ***     LD      HL,(PTRDT)
1241 07A0' FE 04      CP      4
1242 07A2' 28 12      JR      Z,INCPC2
1243 07A4'
1244 07A4' 2A ***     LD      HL,(PTRC) ;CSEG
1245 07A7' 23          INC     HL
1246 07A8' 22 ***     LD      HL,(PTRC) ;HL
1247 07AB' 10 10      JR      INCPC3
1248 07AD'
1249 07AD' 2A ***     INCPC1: LD      HL,(PTRDT) ;DSEG
1250 07B0' 23          INC     HL
1251 07B1' 22 ***     LD      HL,(PTRDT) ;HL
1252 07B4' 10 10      JR      INCPC3
1253 07B6'
1254 07B6' 2A ***     INCPC2: LD      HL,(PTRWK) ;WSEG
1255 07B9' 23          INC     HL
1256 07BA' 22 ***     LD      HL,(PTRWK) ;HL
1257 07BC'
1258 07BC' 2A ***     INCPC3: LD      HL,(PTRFC)
1259 07C0' 23          INC     HL
1260 07C1' 22 ***     LD      HL,(PTRFC) ;HL
1261 07C4' E1          POP     HL
1262 07C5' F1          POP     AF
1263 07C6' C9          RET
1264 07C7'
1265 07C7'
1266 07C7'
1267 07C7' 11 ***     ERR0: LD      DE,MS00
1268 07CA' C0 E5 1F    CALL    MSX
1269 07CB' 3A ***     LD      A,(ITEM)
1270 07CC' C0 C1 1F    CALL    PTRH1
1271 07CD' C0 E2 1F    CALL    LTNL
1272 07CE' C9          RET
1273 07D0'
1274 07D0' E5        ERR1: PUSH    HL
1275 07D0' 11 ***     LD      DE,MS01
1276 07D0' C0 E5 1F    CALL    MSX
1277 07D1' E1          POP     HL
1278 07D1' C0 E2 1F    CALL    PTRH1
1279 07D2' C0 E2 1F    CALL    LTNL
1280 07D3' C9          RET
1281 07D4'
1282 07D4' E5        ERR2: PUSH    HL
1283 07D7' 11 ***     LD      DE,MS02
1284 07D7' C0 E5 1F    CALL    MSX
1285 07D8' E1          POP     HL
1286 07D8' C0 E2 1F    CALL    PTRH1
1287 07D9' C0 E2 1F    CALL    LTNL
1288 07DA' C9          RET
1289 07DB'
1290 07DB' 11 ***     ERR3: LD      DE,MS03
1291 07DB' C0 E5 1F    CALL    MSX
1292 07DB'
1293 07DB' 11 ***     ERR4: LD      DE,MS04
1294 07DB' C0 E5 1F    CALL    MSX
1295 07DB'
1296 07DB' 11 ***     ERR5: LD      DE,MS05
1297 07DB' C0 E5 1F    CALL    MSX
1298 07DB' 11 ***     LD      DE,LBLBUF
1299 07DB' 10 1A      JR      ERROR
1300 07DB'
1301 07DB' 11 ***     ERR6: LD      DE,MS06
1302 07DB' C0 E5 1F    CALL    MSX
1303 07DB' 11 ***     LD      DE,LBLBUF
1304 07DB' 10 1A      JR      ERROR
1305 07DB'
1306 07DB' 11 ***     ERR7: LD      DE,MS07
1307 07DB' 10 1A      JR      ERROR
1308 07DB'
1309 07DB' 11 ***     ERR8: LD      DE,MS08
1310 07DB' 10 1A      JR      ERROR
1311 07DB'
1312 07DB' 11 ***     ERR9: LD      DE,MS09
1313 07DB' 10 1A      JR      ERROR
1314 07DB'
1315 07DB' 11 ***     ERR10: LD      DE,MS10
1316 07DB' 10 1A      JR      ERROR
1317 07DB'
1318 07DB' 11 ***     ERR11: LD      DE,MS11
1319 07DB' 10 1A      JR      ERROR
1320 07DB'
1321 07DB' 11 ***     ERR12: LD      DE,MS12
1322 07DB' 10 1A      JR      ERROR
1323 07DB'
1324 07DB' 11 ***     ERR13: LD      DE,MS13
1325 07DB' 10 1A      JR      ERROR
1326 07DB'
1327 07DB' 11 ***     ERR14: LD      DE,MS14
1328 07DB' 10 1A      JR      ERROR
1329 07DB'
1330 07DB' 11 ***     ERR15: LD      DE,MS15
1331 07DB' 10 1A      JR      ERROR
1332 07DB'
1333 07DB' 11 ***     ERR16: LD      DE,MS16
1334 07DB' 10 1A      JR      ERROR
1335 07DB'
1336 07DB' 11 ***     ERR17: LD      DE,MS17
1337 07DB' 10 1A      JR      ERROR
1338 07DB'
1339 07DB' 11 ***     ERR18: LD      DE,MS18
1340 07DB' 10 1A      JR      ERROR
1341 07DB'
1342 07DB' 11 ***     ERR19: LD      DE,MS19
1343 07DB' 10 1A      JR      ERROR
1344 07DB'
1345 07DB' 11 ***     ERR20: LD      DE,MS20
1346 07DB' 10 1A      JR      ERROR
1347 07DB'
1348 07DB' 11 ***     ERR21: LD      DE,MS21
1349 07DB' 10 1A      JR      ERROR
1350 07DB'
1351 07DB' 11 ***     ERR22: LD      DE,MS22
1352 07DB' 10 1A      JR      ERROR
1353 07DB'
1354 07DB' 11 ***     ERR23: LD      DE,MS23
1355 07DB' 10 1A      JR      ERROR
1356 07DB'
1357 07DB' 11 ***     ERR24: LD      DE,MS24
1358 07DB' 10 1A      JR      ERROR
1359 07DB'
1360 07DB' 11 ***     ERR25: LD      DE,MS25
1361 07DB' 10 1A      JR      ERROR
1362 07DB'
1363 07DB' 11 ***     ERR26: LD      DE,MS26
1364 07DB' 10 1A      JR      ERROR
1365 07DB'
1366 07DB' 11 ***     ERR27: LD      DE,MS27
1367 07DB' 10 1A      JR      ERROR
1368 07DB'
1369 07DB' 11 ***     ERR28: LD      DE,MS28
1370 07DB' 10 1A      JR      ERROR
1371 07DB'
1372 07DB' 11 ***     ERR29: LD      DE,MS29
1373 07DB' 10 1A      JR      ERROR
1374 07DB'
1375 07DB' 11 ***     ERR30: LD      DE,MS30
1376 07DB' 10 1A      JR      ERROR
1377 07DB'
1378 07DB' 11 ***     ERR31: LD      DE,MS31
1379 07DB' 10 1A      JR      ERROR
1380 07DB'
1381 07DB' 11 ***     ERR32: LD      DE,MS32
1382 07DB' 10 1A      JR      ERROR
1383 07DB'
1384 07DB' 11 ***     ERR33: LD      DE,MS33
1385 07DB' 10 1A      JR      ERROR
1386 07DB'
1387 07DB' 11 ***     ERR34: LD      DE,MS34
1388 07DB' 10 1A      JR      ERROR
1389 07DB'
1390 07DB' 11 ***     ERR35: LD      DE,MS35
1391 07DB' 10 1A      JR      ERROR
1392 07DB'
1393 07DB' 11 ***     ERR36: LD      DE,MS36
1394 07DB' 10 1A      JR      ERROR
1395 07DB'
1396 07DB' 11 ***     ERR37: LD      DE,MS37
1397 07DB' 10 1A      JR      ERROR
1398 07DB'
1399 07DB' 11 ***     ERR38: LD      DE,MS38
1400 07DB' 10 1A      JR      ERROR
1401 07DB'
1402 07DB' 11 ***     ERR39: LD      DE,MS39
1403 07DB' 10 1A      JR      ERROR
1404 07DB'
1405 07DB' 11 ***     ERR40: LD      DE,MS40
1406 07DB' 10 1A      JR      ERROR
1407 07DB'
1408 07DB' 11 ***     ERR41: LD      DE,MS41
1409 07DB' 10 1A      JR      ERROR
1410 07DB'
1411 07DB' 11 ***     ERR42: LD      DE,MS42
1412 07DB' 10 1A      JR      ERROR
1413 07DB'
1414 07DB' 11 ***     ERR43: LD      DE,MS43
1415 07DB' 10 1A      JR      ERROR
1416 07DB'
1417 07DB' 11 ***     ERR44: LD      DE,MS44
1418 07DB' 10 1A      JR      ERROR
1419 07DB'
1420 07DB' 11 ***     ERR45: LD      DE,MS45
1421 07DB' 10 1A      JR      ERROR
1422 07DB'
1423 07DB' 11 ***     ERR46: LD      DE,MS46
1424 07DB' 10 1A      JR      ERROR
1425 07DB'
1426 07DB' 11 ***     ERR47: LD      DE,MS47
1427 07DB' 10 1A      JR      ERROR
1428 07DB'
1429 07DB' 11 ***     ERR48: LD      DE,MS48
1430 07DB' 10 1A      JR      ERROR
1431 07DB'
1432 07DB' 11 ***     ERR49: LD      DE,MS49
1433 07DB' 10 1A      JR      ERROR
1434 07DB'
1435 07DB' 11 ***     ERR50: LD      DE,MS50
1436 07DB' 10 1A      JR      ERROR
1437 07DB'
1438 07DB' 11 ***     ERR51: LD      DE,MS51
1439 07DB' 10 1A      JR      ERROR
1440 07DB'
1441 07DB' 11 ***     ERR52: LD      DE,MS52
1442 07DB' 10 1A      JR      ERROR
1443 07DB'
1444 07DB' 11 ***     ERR53: LD      DE,MS53
1445 07DB' 10 1A      JR      ERROR
1446 07DB'
1447 07DB' 11 ***     ERR54: LD      DE,MS54
1448 07DB' 10 1A      JR      ERROR
1449 07DB'
1450 07DB' 11 ***     ERR55: LD      DE,MS55
1451 07DB' 10 1A      JR      ERROR
1452 07DB'
1453 07DB' 11 ***     ERR56: LD      DE,MS56
1454 07DB' 10 1A      JR      ERROR
1455 07DB'
1456 07DB' 11 ***     ERR57: LD      DE,MS57
1457 07DB' 10 1A      JR      ERROR
1458 07DB'
1459 07DB' 11 ***     ERR58: LD      DE,MS58
1460 07DB' 10 1A      JR      ERROR
1461 07DB'
1462 07DB' 11 ***     ERR59: LD      DE,MS59
1463 07DB' 10 1A      JR      ERROR
1464 07DB'
1465 07DB' 11 ***     ERR60: LD      DE,MS60
1466 07DB' 10 1A      JR      ERROR
1467 07DB'
1468 07DB' 11 ***     ERR61: LD      DE,MS61
1469 07DB' 10 1A      JR      ERROR
1470 07DB'
1471 07DB' 11 ***     ERR62: LD      DE,MS62
1472 07DB' 10 1A      JR      ERROR
1473 07DB'
1474 07DB' 11 ***     ERR63: LD      DE,MS63
1475 07DB' 10 1A      JR      ERROR
1476 07DB'
1477 07DB' 11 ***     ERR64: LD      DE,MS64
1478 07DB' 10 1A      JR      ERROR
1479 07DB'
1480 07DB' 11 ***     ERR65: LD      DE,MS65
1481 07DB' 10 1A      JR      ERROR
1482 07DB'
1483 07DB' 11 ***     ERR66: LD      DE,MS66
1484 07DB' 10 1A      JR      ERROR
1485 07DB'
1486 07DB' 11 ***     ERR67: LD      DE,MS67
1487 07DB' 10 1A      JR      ERROR
1488 07DB'
1489 07DB' 11 ***     ERR68: LD      DE,MS68
1490 07DB' 10 1A      JR      ERROR
1491 07DB'
1492 07DB' 11 ***     ERR69: LD      DE,MS69
1493 07DB' 10 1A      JR      ERROR
1494 07DB'
1495 07DB' 11 ***     ERR70: LD      DE,MS70
1496 07DB' 10 1A      JR      ERROR
1497 07DB'
1498 07DB' 11 ***     ERR71: LD      DE,MS71
1499 07DB' 10 1A      JR      ERROR
1500 07DB'
1501 07DB' 11 ***     ERR72: LD      DE,MS72
1502 07DB' 10 1A      JR      ERROR
1503 07DB'
1504 07DB' 11 ***     ERR73: LD      DE,MS73
1505 07DB' 10 1A      JR      ERROR
1506 07DB'
1507 07DB' 11 ***     ERR74: LD      DE,MS74
1508 07DB' 10 1A      JR      ERROR
1509 07DB'
1510 07DB' 11 ***     ERR75: LD      DE,MS75
1511 07DB' 10 1A      JR      ERROR
1512 07DB'
1513 07DB' 11 ***     ERR76: LD      DE,MS76
1514 07DB' 10 1A      JR      ERROR
1515 07DB'
1516 07DB' 11 ***     ERR77: LD      DE,MS77
1517 07DB' 10 1A      JR      ERROR
1518 07DB'
1519 07DB' 11 ***     ERR78: LD      DE,MS78
1520 07DB' 10 1A      JR      ERROR
1521 07DB'
1522 07DB' 11 ***     ERR79: LD      DE,MS79
1523 07DB' 10 1A      JR      ERROR
1524 07DB'
1525 07DB' 11 ***     ERR80: LD      DE,MS80
1526 07DB' 10 1A      JR      ERROR
1527 07DB'
1528 07DB' 11 ***     ERR81: LD      DE,MS81
1529 07DB' 10 1A      JR      ERROR
1530 07DB'
1531 07DB' 11 ***     ERR82: LD      DE,MS82
1532 07DB' 10 1A      JR      ERROR
1533 07DB'
1534 07DB' 11 ***     ERR83: LD      DE,MS83
1535 07DB' 10 1A      JR      ERROR
1536 07DB'
1537 07DB' 11 ***     ERR84: LD      DE,MS84
1538 07DB' 10 1A      JR      ERROR
1539 07DB'
1540 07DB' 11 ***     ERR85: LD      DE,MS85
1541 07DB' 10 1A      JR      ERROR
1542 07DB'
1543 07DB' 11 ***     ERR86: LD      DE,MS86
1544 07DB' 10 1A      JR      ERROR
1545 07DB'
1546 07DB' 11 ***     ERR87: LD      DE,MS87
1547 07DB' 10 1A      JR      ERROR
1548 07DB'
1549 07DB' 11 ***     ERR88: LD      DE,MS88
1550 07DB' 10 1A      JR      ERROR
1551 07DB'
1552 07DB' 11 ***     ERR89: LD      DE,MS89
1553 07DB' 10 1A      JR      ERROR
1554 07DB'
1555 07DB' 11 ***     ERR90: LD      DE,MS90
1556 07DB' 10 1A      JR      ERROR
1557 07DB'
1558 07DB' 11 ***     ERR91: LD      DE,MS91
1559 07DB' 10 1A      JR      ERROR
1560 07DB'
1561 07DB' 11 ***     ERR92: LD      DE,MS92
1562 07DB' 10 1A      JR      ERROR
1563 07DB'
1564 07DB' 11 ***     ERR93: LD      DE,MS93
1565 07DB' 10 1A      JR      ERROR
1566 07DB'
1567 07DB' 11 ***     ERR94: LD      DE,MS94
1568 07DB' 10 1A      JR      ERROR
1569 07DB'
1570 07DB' 11 ***     ERR95: LD      DE,MS95
1571 07DB' 10 1A      JR      ERROR
1572 07DB'
1573 07DB' 11 ***     ERR96: LD      DE,MS96
1574 07DB' 10 1A      JR      ERROR
1575 07DB'
1576 07DB' 11 ***     ERR97: LD      DE,MS97
1577 07DB' 10 1A      JR      ERROR
1578 07DB'
1579 07DB' 11 ***     ERR98: LD      DE,MS98
1580 07DB' 10 1A      JR      ERROR
1581 07DB'
1582 07DB' 11 ***     ERR99: LD      DE,MS99
1583 07DB' 10 1A      JR      ERROR
1584 07DB'
1585 07DB' 11 ***     ERR100: LD      DE,MS100
1586 07DB' 10 1A      JR      ERROR
1587 07DB'
1588 07DB' 11 ***     ERR101: LD      DE,MS101
1589 07DB' 10 1A      JR      ERROR
1590 07DB'
1591 07DB' 11 ***     ERR102: LD      DE,MS102
1592 07DB' 10 1A      JR      ERROR
1593 07DB'
1594 07DB' 11 ***     ERR103: LD      DE,MS103
1595 07DB' 10 1A      JR      ERROR
1596 07DB'
1597 07DB' 11 ***     ERR104: LD      DE,MS104
1598 07DB' 10 1A      JR      ERROR
1599 07DB'
1600 07DB' 11 ***     ERR105: LD      DE,MS105
1601 07DB' 10 1A      JR      ERROR
1602 07DB'
1603 07DB' 11 ***     ERR106: LD      DE,MS106
1604 07DB' 10 1A      JR      ERROR
1605 07DB'
1606 07DB' 11 ***     ERR107: LD      DE,MS107
1607 07DB' 10 1A      JR      ERROR
1608 07DB'
1609 07DB' 11 ***     ERR108: LD      DE,MS108
1610 07DB' 10 1A      JR      ERROR
1611 07DB'
1612 07DB' 11 ***     ERR109: LD      DE,MS109
1613 07DB' 10 1A      JR      ERROR
1614 07DB'
1615 07DB' 11 ***     ERR110: LD      DE,MS110
1616 07DB' 10 1A      JR      ERROR
1617 07DB'
1618 07DB' 11 ***     ERR111: LD      DE,MS111
1619 07DB' 10 1A      JR      ERROR
1620 07DB'
1621 07DB' 11 ***     ERR112: LD      DE,MS112
1622 07DB' 10 1A      JR      ERROR
1623 07DB'
1624 07DB' 11 ***     ERR113: LD      DE,MS113
1625 07DB' 10 1A      JR      ERROR
1626 07DB'
1627 07DB' 11 ***     ERR114: LD      DE,MS114
1628 07DB' 10 1A      JR      ERROR
1629 07DB'
1630 07DB' 11 ***     ERR115: LD      DE,MS115
1631 07DB' 10 1A      JR      ERROR
1632 07DB'
1633 07DB' 11 ***     ERR116: LD      DE,MS116
1634 07DB' 10 1A      JR      ERROR
1635 07DB'
1636 07DB' 11 ***     ERR117: LD      DE,MS117
1637 07DB' 10 1A      JR      ERROR
1638 07DB'
1639 07DB' 11 ***     ERR118: LD      DE,MS118
1640 07DB' 10 1A      JR      ERROR
1641 07DB'
1642 07DB' 11 ***     ERR119: LD      DE,MS119
1643 07DB' 10 1A      JR      ERROR
1644 07DB'
1645 07DB' 11 ***     ERR120: LD      DE,MS120
1646 07DB' 10 1A      JR      ERROR
1647 07DB'
1648 07DB' 11 ***     ERR121: LD      DE,MS121
1649 07DB' 10 1A      JR      ERROR
1650 07DB'
1651 07DB' 11 ***     ERR122: LD      DE,MS122
1652 07DB' 10 1A      JR      ERROR
1653 07DB'
1654 07DB' 11 ***     ERR123: LD      DE,MS123
1655 07DB' 10 1A      JR      ERROR
1656 07DB'
1657 07DB' 11 ***     ERR124: LD      DE,MS124
1658 07DB' 10 1A      JR      ERROR
1659 07DB'
1660 07DB' 11 ***     ERR125: LD      DE,MS125
1661 07DB' 10 1A      JR      ERROR
1662 07DB'
1663 07DB' 11 ***     ERR126: LD      DE,MS126
1664 07DB' 10 1A      JR      ERROR
1665 07DB'
1666 07DB' 11 ***     ERR127: LD      DE,MS127
1667 07DB' 10 1A      JR      ERROR
1668 07DB'
1669 07DB' 11 ***     ERR128: LD      DE,MS128
1670 07DB' 10 1A      JR      ERROR
1671 07DB'
1672 07DB' 11 ***     ERR129: LD      DE,MS129
1673 07DB' 10 1A      JR      ERROR
1674 07DB'
1675 07DB' 11 ***     ERR130: LD      DE,MS130
1676 07DB' 10 1A      JR      ERROR
1677 07DB'
1678 07DB' 11 ***     ERR131: LD      DE,MS131
1679 07DB' 10 1A      JR      ERROR
1680 07DB'
1681 07DB' 11 ***     ERR132: LD      DE,MS132
1682 07DB' 10 1A      JR      ERROR
1683 07DB'
1684 07DB' 11 ***     ERR133: LD      DE,MS133
1685 07DB' 10 1A      JR      ERROR
1686 07DB'
1687 07DB' 11 ***     ERR134: LD      DE,MS134
1688 07DB' 10 1A      JR      ERROR
1689 07DB'
1690 07DB' 11 ***     ERR135: LD      DE,MS135
1691 07DB' 10 1A      JR      ERROR
1692 07DB'
1693 07DB' 11 ***     ERR136: LD      DE,MS136
1694 07DB' 10 1A      JR      ERROR
1695 07DB'
1696 07DB' 11 ***     ERR137: LD      DE,MS137
1697 07DB' 10 1A      JR      ERROR
1698 07DB'
1699 07DB' 11 ***     ERR138: LD      DE,MS138
1700 07DB' 10 1A      JR      ERROR
1701 07DB'
1702 07DB' 11 ***     ERR139: LD      DE,MS139
1703 07DB' 10 1A      JR      ERROR
1704 07DB'
1705 07DB' 11 ***     ERR140: LD      DE,MS140
1706 07DB' 10 1A      JR      ERROR
1707 07DB'
1708 07DB' 11 ***     ERR141: LD      DE,MS141
1709 07DB' 10 1A      JR      ERROR
1710 07DB'
1711 07DB' 11 ***     ERR142: LD      DE,MS142
1712 07DB' 10 1A      JR      ERROR
1713 07DB'
1714 07DB' 11 ***     ERR143: LD      DE,MS143
1715 07DB' 10 1A      JR      ERROR
1716 07DB'
1717 07DB' 11 ***     ERR144: LD      DE,MS144
1718 07DB' 10 1A      JR      ERROR
1719 07DB'
1720 07DB' 11 ***     ERR145: LD      DE,MS145
1721 07DB' 10 1A      JR      ERROR
1722 07DB'
1723 07DB' 11 ***     ERR146: LD      DE,MS146
1724 07DB' 10 1A      JR      ERROR
1725 07DB'
1726 07DB' 11 ***     ERR147: LD      DE,MS147
1727 07DB' 10 1A      JR      ERROR
1728 07DB'
1729 07DB' 11 ***     ERR148: LD      DE,MS148
1730 07DB' 10 1A      JR      ERROR
1731 07DB'
1732 07DB' 11 ***     ERR149: LD      DE,MS149
1733 07DB' 10 1A      JR      ERROR
1734 07DB'
1735 07DB' 11 ***     ERR150: LD      DE,MS150
1736 07DB' 10 1A      JR      ERROR
1737 07DB'
1738 07DB' 11 ***     ERR151: LD      DE,MS151
1739 07DB' 10 1A      JR      ERROR
1740 07DB'
1741 07DB' 11 ***     ERR152: LD      DE,MS152
1742 07DB' 10 1A      JR      ERROR
1743 07DB'
1744 07DB' 11 ***     ERR153: LD      DE,MS153
1745 07DB' 10 1A      JR      ERROR
1746 07DB'
1747 07DB' 11 ***     ERR154: LD      DE,MS154
1748 07DB' 10 1A      JR      ERROR
1749 07DB'
1750 07DB' 11 ***     ERR155: LD      DE,MS155
1751 07DB' 10 1A      JR      ERROR
1752 07DB'
1753 07DB' 11 ***     ERR156: LD      DE,MS156
1754 07DB' 10 1A      JR      ERROR
1755 07DB'
1756 07DB' 11 ***     ERR157: LD      DE,MS157
1757 07DB' 10 1A      JR      ERROR
1758 07DB'
1759 07DB' 11 ***     ERR158: LD      DE,MS158
1760 07DB' 10 1A      JR      ERROR
1761 07DB'
1762 07DB' 11 ***     ERR159: LD      DE,MS159
1763 07DB' 10 1A      JR      ERROR
1764 07DB'
1765 07DB' 11 ***     ERR160: LD      DE,MS160
1766 07DB' 10 1A      JR      ERROR
1767 07DB'
1768 07DB' 11 ***     ERR161: LD      DE,MS161
1769 07DB' 10 1A      JR      ERROR
1770 07DB'
1771 07DB' 11 ***     ERR162: LD      DE,MS162
1772 07DB' 10 1A      JR      ERROR
1773 07DB'
1774 07DB' 11 ***     ERR163: LD      DE,MS163
1775 07DB' 10 1A      JR      ERROR
1776 07DB'
1777 07DB' 11 ***     ERR164: LD      DE,MS164
1778 07DB' 10 1A      JR      ERROR
1779 07DB'
1780 07DB' 11 ***     ERR165: LD      DE,MS165
1781 07DB' 10 1A      JR      ERROR
1782 07DB'
1783 07DB' 11 ***     ERR166: LD      DE,MS166
1784 07DB' 10 1A      JR      ERROR
1785 07DB'
1786 07DB' 11 ***     ERR167: LD      DE,MS167
1787 07DB' 10 1A      JR      ERROR
1788 07DB'
1789 07DB' 11 ***     ERR168: LD      DE,MS168
1790 07DB' 10 1A      JR      ERROR
1791 07DB'
1792 07DB' 11 ***     ERR169: LD      DE,MS169
1793 07DB' 10 1A      JR      ERROR
1794 07DB'
1795 07DB' 11 ***     ERR170: LD      DE,MS170
1796 07DB' 10 1A      JR      ERROR
1797 07DB'
1798 07DB' 11 ***     ERR171: LD      DE,MS171
1799 07DB' 10 1A      JR      ERROR
1800 07DB'
1801 07DB' 11 ***     ERR172: LD      DE,MS172
1802 07DB' 10 1A      JR      ERROR
1803 07DB'
1804 07DB' 11 ***     ERR173: LD      DE,MS173
1805 07DB' 10 1A      JR      ERROR
1806 07DB'
1807 07DB' 11 ***     ERR174: LD      DE,MS174
1808 07DB' 10 1A      JR      ERROR
1809 07DB'
1810 07DB' 11 ***     ERR175: LD      DE,MS175
1811 07DB' 10 1A      JR      ERROR
1812 07DB'
1813 07DB' 11 ***     ERR176: LD      DE,MS176
1814 07DB' 10 1A      JR      ERROR
1815 07DB'
1816 07DB' 11 ***     ERR177: LD      DE,MS177
1817 07DB' 10 1A      JR      ERROR
1818 07DB'
1819 07DB' 11 ***     ERR178: LD      DE,MS178
1820 07DB' 10 1A      JR      ERROR
1821 07DB'
1822 07DB' 11 ***     ERR179: LD      DE,MS179
1823 07DB' 10 1A      JR      ERROR
1824 07DB'
1825 07DB' 11 ***     ERR180: LD      DE,MS180
1826 07DB' 10 1A      JR      ERROR
1827 07DB'
1828 07DB' 11 ***     ERR181: LD      DE,MS181
1829 07DB' 10 1A      JR      ERROR
1830 07DB'
1831 07DB' 11 ***     ERR182: LD      DE,MS182
1832 07DB' 10 1A      JR      ERROR
1833 07DB'
1834 07DB' 11 ***     ERR183: LD      DE,MS183
1835 07DB' 10 1A      JR      ERROR
1836 07DB'
1837 07DB' 11 ***     ERR184: LD      DE,MS184
1838 07DB' 10 1A      JR      ERROR
1839 07DB'
1840 07DB' 11 ***     ERR185: LD      DE,MS185
1841 07DB' 10 1A      JR      ERROR
1842 07DB'
1843 07DB' 11 ***     ERR186: LD      DE,MS186
1844 07DB' 10 1A      JR      ERROR
1845 07DB'
1846 07DB' 11 ***     ERR187: LD      DE,MS187
1847 07DB' 10 1A      JR      ERROR
1848 07DB'
1849 07DB' 11 ***     ERR188: LD      DE,MS188
1850 07DB' 10 1A      JR      ERROR
1851 07DB'
1852 07DB' 11 ***     ERR189: LD      DE,MS189
1853 07DB' 10 1A      JR      ERROR
1854 07DB'
1855 07DB' 11 ***     ERR190: LD      DE,MS190
1856 07DB' 10 1A      JR      ERROR
1857 07DB'
1858 07DB' 11 ***     ERR191: LD      DE,MS191
1859 07DB' 10 1A      JR      ERROR
1860 07DB'
1861 07DB' 11 ***     ERR192: LD      DE,MS192
1862 07DB' 10 1A      JR      ERROR
1863 07DB'
1864 07DB' 11 ***     ERR193: LD      DE,MS193
1865 07DB' 10 1A      JR      ERROR
1866 07DB'
1867 07DB' 11 ***     ERR194: LD      DE,MS194
1868 07DB' 10 1A      JR      ERROR
1869 07DB'
1870 07DB' 11 ***     ERR195: LD      DE,MS195
1871 07DB' 10 1A
```


156 Oh! X 1990.8. ▶編集部の住所が変わってしまった。せっかく全日本プロレス武道館大会の帰りに編集室に結果報告に行こうと思っていたのに。高輪かあ。慶応が明治学院大学でも受験しようかなあ。
津田 典秀(20)千葉県

```

611 0339" CB 3C SKL H
612 0341" CB 1D SKL L ;HL/2
613 0343" CB 3C SKL H
614 0345" CB 1D SKL L ;HL/4
615 0347" CB 3C SKL H
616 0349" CB 1D SKL L ;HL/6
617 034B" CB 3C SKL H
618 034D" CB 1D SKL L ;HL/16
619 034F" J0 LD A,L
620 0350" J1 POP HL
621 0351" C9 RET
622 0352" ;
623 0353" ; Works
624 0354" ;
625 0355" ;
626 0356" ;
627 0357" DSEG
628 0000"
629 0001" PCN_ADDR:DS 2 ;File Control Block Address
630 0002" LST_DSK:DS 1 ;Disk driven last
631 0003"
632 0004"
633 0005" INF_SIZE EQU 37H
634 0006" DS INF_SIZE ;work area for Input-f
635 0007" wkout:: DS INF_SIZE ;work area for Output-f
636 0008"
637 0009"
638 0010" FILE_SF:DS 12H
639 0011" FLSTSIZE: DS 2 ;File Size.
640 0012"

```

```

639 0005" FLDADR:DS 2 ;Start Address.
640 0007" FLDADR:DS 2 ;Exec Address.
641 0009" DS 6
642 000F" FSTCLST:DS 1 ;First Cluster.
643 0010" DS 1
644 0011" TRCLST:DS 16H ;Cluster table.
645 0012" FLDSC: DS 1 ;The Login disk.
646 0013" FLPT: DS 1 ;The FILE Pointer.
647 0014" DEBUF: DS 2 ;Record No. Which Have The DI
648 0015" HLBUFF: DS 2 ;Address Where On INFORMATION
649 0016" RC: DS 1 ;The Number of Records The File
650 0017" have.
651 0018"
652 0019" END

```

```

6 cnduf EQU 5000H ;- 5FFF
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H
6 cnduf EQU 5000H

```

リスト5

```

1: Header File For WLC
2: CSEG 3000H-
3: DSEG 4500H-
4:
5: LBLMAX EQU 1000H

```

リスト6

```

1: Header File For W2D
2: CSEG 3000H-
3: DSEG 6000H-
4:
5:
6: RDBUF EQU 0000H
7: WRBUF1 EQU 0000H
8: WRBUF2 EQU 0000H
9:

```

全機種共通システムインデックス

■85年 6月号

序論 共通化の試み

第1部 S-OS"MACE"

第2部 Lisp-85インタプリタ

第3部 チェックサムプログラム

■85年 7月号

第4部 マシン語プログラム開発入門

第5部 エディタアセンブラZEDA

第6部 デバッグツールZAID

■85年 8月号

第7部 ゲーム開発パッケージBEMS

第8部 ソースジェネレータZING

■85年 9月号

インタラプト S-OS番外地

第9部 マシン語入力ツールMACINTO-S

第10部 Lisp-85入門(1)

■85年 10月号

第11部 仮想マシンCAP-X85

連載 Lisp-85入門(2)

■85年 11月号

連載 Lisp-85入門(3)

■85年 12月号

第12部 Prolog-85発表

■86年 1月号

第13部 リロケータブルのお話

第14部 FM音源サウンドエディタ

■86年 2月号

第15部 S-OS"SWORD"

第16部 Prolog-85入門(1)

■86年 3月号

第17部 magiFORTH発表

連載 Prolog-85入門(2)

■86年 4月号

第18部 思考ゲームJEWEL

第19部 LIFE GAME

連載 基礎からのmagiFORTH

連載 Prolog-85入門(3)

■86年 5月号

第20部 スクリーンエディタE-MATE

連載 実戦演習magiFORTH

■86年 6月号

第21部 Z80TRACER

第22部 magiFORTH TRACER

第23部 ディスクダンプ&エディタ

第24部 "SWORD" 2000 QD

連載 対話で学ぶ magiFORTH

特別付録 PC-8801版 S-OS"SWORD"

■86年 7月号

第25部 FM音源ミュージックシステム

付録 FM音源ボードの製作

連載 計算力アップのmagiFORTH

特別付録 SMC-777版 S-OS"SWORD"

■86年 8月号

第26部 対局五目並べ

第27部 MZ-2500版 S-OS"SWORD"

■86年 9月号

第28部 FuzzyBASIC 発表

連載 明日に向かって magiFORTH

■86年 10月号

第29部 ちょっと便利な拡張プログラム

第30部 ディスクモニタ DREAM

第31部 FuzzyBASIC 料理法<1>

■86年 11月号

第32部 バズルゲーム HOTTAN

第33部 MAZE in MAZE

連載 FuzzyBASIC 料理法<2>

■86年 12月号

第34部 CASL & COMET

連載 FuzzyBASIC 料理法<3>

■87年 1月号

第35部 マシン語入力ツールMACINTO-C

連載 FuzzyBASIC 料理法<4>

■87年 2月号

第36部 アドベンチャーゲーム MARMALADE

第37部 テキアベ作成ツール CONTEX

■87年 3月号

第38部 魔法使いはアニメが大好き

第39部 アニメーションツール MAGE

付録 "SWORD" 再掲載と MAGIC の標準化

■87年 4月号

第40部 INVADER GAME

第41部 TANGERINE

■87年 5月号

第42部 S-OS"SWORD" 変身セット

第43部 MZ-700用 "SWORD" を QD 対応に

■87年 6月号

インタラプト コンパイル物語

第44部 FuzzyBASIC コンパイラ

第45部 エディタアセンブラ ZEDA-3

■87年 7月号

第46部 STORY MASTER

■87年 8月号

第47部 バズルゲーム 碁石拾い

第48部 漢字出力パッケージ JACKWRITE

特別付録 FM-7/77版 S-OS"SWORD"

■87年 9月号

第49部 リロケータブル逆アセンブラ Inside-R

特別付録 PC-8001/8801 版 S-OS"SWORD"

■87年 10月号

第50部 tiny CORE WARS

第51部 FuzzyBASIC コンパイラの拡張

第52部 Xturbo 版 S-OS"SWORD"

■87年 11月号

序論 神話のなかのマイクロコンピュータ

付録 S-OS の仲間たち

第53部 もうひとつの FuzzyBASIC 入門

第54部 ファイルアロケータ&ローダ

インタラプト S-OS はこちら集中治療室

■87年 12月号

第55部 BACK GAMMON

第56部 タートルグラフィックパッケージTURTLE

第57部 Xturbo 版 "SWORD" アフターケア

ラインプリントルーチン

特別付録 PASOPIA7 版 S-OS"SWORD"

■88年 1月号

第58部 FuzzyBASIC コンパイラ・奥村版

付録 石上版コンパイラ拡張部の修正

■88年 2月号

第59部 シューティングゲーム ELFES

■88年 3月号

第60部 構造型コンパイラ言語 SLANG

■88年 4月号

第61部 デバッグツール TRADE

第62部 シミュレーションウーゲーム WALRUS

■88年 5月号

第63部 シューティングゲーム ELFES II

第64部 地底最大の作戦

■88年 6月号

第65部 構造化言語 SLANG 入門(1)

第66部 Lisp-85 用 NAMPA シミュレーション

■88年 7月号

第67部 マルチウィンドウドライバ MW-1

連載 構造化言語 SLANG 入門(2)

■88年 8月号

第68部 マルチウィンドウエディタ WINER

■88年 9月号

第69部 超小型エディタ TED-750

第70部 アフターケア WINER の拡張

■88年 10月号

第71部 SLANG 用ファイル入出力ライブラリ

第72部 シューティングゲーム MANKAI

■88年 11月号

第73部 シューティングゲーム ELFES IV

■88年 12月号

第74部 ソースジェネレータ SOURCERY

■89年 1月号

第75部 バズルゲーム LAST ONE

第76部 ブロックゲーム FLICK

■89年 2月号

第77部 高速エディタアセンブラ REDA

特別付録 X1版 S-OS"SWORD"<再掲載>

■89年 3月号

第78部 Z80用浮動小数点演算パッケージSOROBAN

■89年 4月号

第79部 SLANG 用実数演算ライブラリ

■89年 5月号

第80部 ソースジェネレータ RING

■89年 6月号

第81部 超小型コンパイラ TTC

■89年 7月号

第82部 TTC用バズルゲーム TICBAN

■89年 8月号

第83部 CP/M 用 ファイルコンバータ

■89年 9月号

第84部 生物進化シミュレーションBUGS

■89年 10月号

第85部 小型インタプリタ言語 TTI

■89年 11月号

第86部 TTI用バズルゲーム PUSH BON!

■89年 12月号

第87部 SLANG用リダイレクションライブラリ

DIO. LIB

■90年 1月号

第88部 SLANG用ゲーム WORM KUN

特別付録 再掲載SLANGコンパイラ

■90年 2月号

第89部 超小型コンパイラTTC++

■90年 3月号

第90部 超多機能アセンブラOHM-Z80

■90年 4月号

第91部 ファジィコンピュータシミュレーション-MY

■90年 5月号

第92部 インタプリタ言語STACK

■90年 6月号

第93部 リロケータブルフォーマットの取り決め

第94部 STACK用ゲーム SQUASH !

第95部 X68000対応S-OS"SWORD"

特別付録 PC-286対応S-OS"SWORD"

■90年 7月号

第96部 リロケータブルアセンブラWZD

* 以上のアプリケーションは、基本システムである

S-OS "MACE" または S-OS"SWORD" がないと動作

しませんのでご注意ください。

▶ 7月号は何とんでもなく133ページの予告、西川さんのFAXの絵が一番面白かった。「勝てるものなら勝ってみろ」ふー、相当自信があるんだなあ。結果が楽しみだ。しかし、「あらかじめご了承ください」攻撃はやめなさい!?

功刀 和久(21)埼玉県

人工知能の冒険

完全な真空

毛色の変った本を出すとして有名な国書刊行会から出されている『完全な真空』という、おかしな本をぜひとも皆さんに紹介しようと思います。著者はスタニスワフ・レムでして、タルコフスキーの撮った「惑星ソラリス」という映画の原作者として有名です。レムはSF作家としてきわめて有名であり、「最高のSF作家」とさえ呼ばれているそうです。

この『完全な真空』は、本当はこの世に存在しない本を、まるで存在するかのように出版社や作者名まででっち上げたうえ、それらの本それぞれに対する書評をまた自分で書いているというものです。ひとことでいえば、架空書評集というところでしょう。

全部で16冊の架空の書物が取り上げられているのですが、おもしろいものとおもしろくないものの差がきわめて大きく感じられました。あまり興味がもてなかったのが、『親衛隊少将ルイ16世』や『白痴』のように、なにかスケールの異常に大きい大作の概略を示したようなものです。逆に、3度も4度も読んでしまったのが、「最高のSF作家」こそが書きうるといようなものです。

実在する1冊の本

正確にいうならば、この本に収録されている16冊の架空書物のうち、先頭に取り上げられている1冊だけは実在します。それは、この本『完全な真空』自体です。そこでは、まるで別の人が書いたように、「レム氏は……」などとしらばっくれて書いています。しかもさらに、その文章の中で、その文章そのものを引用することまで行い、一層混乱の度をわざと高めています。

書き上げてもいない本を作り出し、それを今度は評論家の立場で好き勝手に批評し、そうしてできた本をまた同じ本の中で批評するとは、いってみればなんとも書きとしてぜいたくなことをやっているのだろう

と思わず感じてしまいます。

・このように書評集の中でその書評集自体を取り上げるというのは、「再帰呼び出し」(リカーシブコール)を思い起こさせます。この例に見られるように、再帰呼び出し的なことは単にプログラムの中の関数の呼び出し方だけに限定された話ではありません。ネーミングの中に見られるごく簡単な例を示しましょう。UNIXオペレーティングシステムの発展版にそのスペルを引っ繰り返したXINU(ジーニュ)というのがありますが、これは次の文章の頭文字をとったものだそうです。

“XINU Is Not Unix.”

研究室にあるUNIXマシンのひとつ(CPUはSPARC)の名前を、SPARCを引っ繰り返したCRAPSとしているのですが、その名前の由来も無理やりこのXINUのように説明するならば、

“CRAPS Runs A Processor Sparc.”
(CRAPSはSparcプロセッサを駆動する)とでもいえばよいのでしょう。

存在しえない小説

『完全な真空』の中で取り上げられている仮想小説のうちのひとつに「とどのつまりは何も無し」というものがあります。この小説についてここで紹介し、読者の皆さんにああこういう小説なのかとわかってもらうことは、きわめてむずかしいことと思われます。第一、僕自身どう考えても、このような小説がどのように存在し得るか、あまり想像がつかないからです。

まあとにかく、この小説の内容を紹介することにしましょう(無駄とわかっていても)。この小説の内容はないのです。といっても、真っ白な紙が並んでいるのではなく、しっかりと文章が並んでいるのです(もちろん、「何もない」と1000回書かれているわけでもありません)。しかし、何も語ってはいないのです。

冒頭の文は「列車は着かなかった」となっています。そして、「誰か」が現れなかったあと、語りは非人称のまま、時は春で

もなく夏でもなく、無重力空間における愛されない女に関する考察によって第1章は閉じられます。

その後、この本に関する記述は抽象度を増します。「虚無の穴が不気味に大きくなってゆく」「思考しないことの流れ」「テキストはわれわれの所有していたものを次々と奪い取っていく」……。作品の最後ではもうこれ以上作品が続くかという疑念が沸き起こってきます。

そして、ついには「存在しないこと」は否定として存在することさえやめてしまうのです。文章の意味が失われると残るのは構文のみです。しかしその文法装置さえしまいには空中分解してしまい、文章の途中、単語の途中でついにこの小説は終わってしまうのです(とまあちょっただけ書いてみましたが、やはり徒労に終わったのでしょうね?)。

でも、実際には存在しえない小説を仮想することこそ、この本の真価といえるでしょう。しかもなぜこのような小説がこの世に存在するかという意味づけもしっかりとなされています。要するに、小説家が誠実さを究極にまで追求したときに必然的に生まれる小説は、まさにこのようなものであるということです。小説家はありもしないことを書かなくてはならないのですが、もしそのような行為に良心の呵責を感じるような小説家が万一存在したならば、彼の取るべき道は2つだけ、筆を折るか、あるいは「とどのつまりは何も無い」小説を書くかということなのです。

このような小説を書く小説家の誠実さについて論じながらレムは、「私はそのような意味での誠実さからはいちばん遠いのだ」と含み笑っていることでしょう。小説家が誠実さを求めることは、レムの行っている「ありもしない小説をでっち上げる」行為とちょうど正反対であるからなのです。

ところで、この世に存在しない小説の書評をした本を取り上げて、それをまた書評している僕自身の誠実さはいったいどうな

っているのでしょうか？ まあ、この『完全な真空』という本が存在しないのならば、それこそ賞賛に値するほどの不誠実さともいえるでしょうが、僕はまだまだ……。

知能の相対化

既成のとりわかれた概念に対する鋭い風刺の効いた疑問は、この本のいろいろなところに見られます。「誤謬としての文化」では、まず、「文化は生物が生き残る邪魔にもならなければ、助けにもならないものである」という考えを否定します（これはまあ普通の主張といえましょう）。ところが次に主張されるのはきわめて刺激的な考えです。「文化というものは、自ら作り出した宗教、慣習、法、禁止、命令を通じて作用することにより、不十分なものを理想に、マイナスをプラスに、欠点や欠陥のあるものを完璧なものに作り変えるのだ」というのです。

あるいは別の書評では、知能というものに関して、人間の知能の絶対性というものに強い懐疑を示します。そしてこれは、「完全な真空」以外の彼の書物にも見られる一貫した態度のようです。人工知能という言葉は、最近ではごく当たり前に使われる言葉になってきたのですが、その際、知能は人間の頭脳こそが唯一もっているものであるということは、当然のこと、暗黙の了解事項であるように僕には感じられます。「ソラリス」のテーマ自体がそうであったように、レムはいつも人間のもっているものが知能として絶対唯一であるということへの疑問を提示しています。それどころか、この本を読むと人間の知能などは偶然の産物なのだという声さえ、きわめて皮肉的かつ間接的ではありますが、聞こえてきます。

この本が書かれたのはなんと1971年です（日本語訳が出たのは1989年）。その後10年くらいたって、いわゆるサイバーパンクといわれる新しい潮流が生まれて、人間の脳の神経細胞のクローズアップ、たとえば、直接、神経細胞をメディアとしてコミュニ

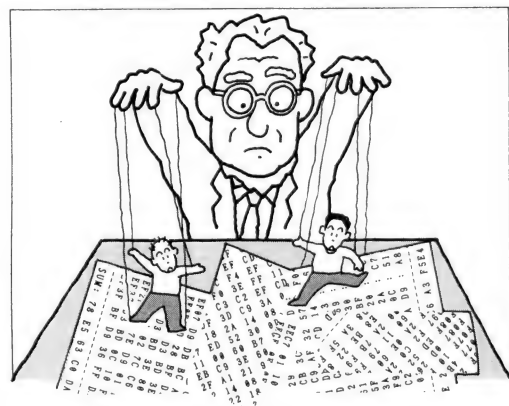
ケーションするという考えなどが生まれたわけです。次に紹介する架空書物評などを読むと、この本が今から20年も前に書かれたとは信じられない気がします。

「我は下僕ならずや」では、現実世界からはまったく切り離された神経細胞における電気パルスの伝達のみ構成される世界というものを、さらに独立させ、純粹化した世界を描いています。キーワードはパーソネティクス（理性ある生物の人工生産）なのだそうです。そのような世界を小説として描いているのではなく、実際にそのような世界を研究室の計算機内に作り上げたドブ教授がこの架空小説の著者なのです。

もうひとつ別の人工知能

「我は下僕ならずや」で記述されている世界といっても、完全に計算機の中の閉じた世界であり、まったく数学的に作り上げられたもののなのです。しかしそこには「住人」がいるのです。なぜそこに、知能をもつ生命体が住んでいるとみなせるかというところがミソであり、えんえんとページが費やされています。いわゆるシミュレーションのようにも思えるのですが、シミュレーションではありません、実体なのです。現実の物理的空間がないところになぜ知的生命体が想定できるのでしょうか？

このことについては、人間が住んでいるこの世というものが実は偶然の産物であり、数学的な世界の中にも、人間世界とまったく同じような現象が起こりうるということを執拗に述べています。偶然この世は3次元なのですが、彼ら「住人」の住む数学的世界ではそれらが任意に（ドブ教授によって）設定することができるのです。時間の進み具合も設定できます。ある種の具体化を遂げた数学は、完全に実体をもたぬほどに精神化した知性の生活空間となりえたのです。さらにレムは、この世や人間に特有なささまざまな概念、たとえば、意



識、言語、進化などに関して、その脆弱性（もろくて弱いということ）を追求し、そして計算機の中に閉じ込められた世界でも同様の概念が存在するというのです。

この架空小説が最大に盛り上がるのは、「住人」たちの、創造主（つまりこの架空小説を書いているドブ教授）に関する議論です。何人かの「住人」たちが、いったい創造主はいるのかいないのか、いるのならば、今の我々とどういう関係にあるのかということをは話し合うのです。おもしろいのは、彼らの世界を述べているようで、いつの間にか、実は我々人間自身の問題と完全にオーバーラップしてくることです。

レムは実は計算機の中の人間が作り出した世界に生きる知的生命体を描きながら、実は、我々もまた上のレベルにある何者か（創造主）に操られているというような循環をも同時に描いているのでしょう。

知能機械といっても、人がもっているような知能だけを相手にしているのではもう古いのかもしれませんが。50年先、500年先をにらんで生きていく人は、ソラリスの海やサイバーパンクやパーソネティクスまでも包括したものとして、知能というものをイメージしていかねばならないのでしょう。

というわけで、本連載でも、総力を込めてというか、脱線しまくってというか、次回には、毛色の全然違う未知の領域に踏み込もうと思います。タイトルは、「超能力大実験：ここにも超能力者が！（仮称）」です。（こりゃとんだことになりそうだと感じつつ）来月をお楽しみに！

猫とコンピュータ サーチャーでござる

Takazawa Kyoko

高沢 恭子



あれってどこ置いたんだっけ？ っていうときは、自分がそれを置きそうなところや隠れそうなところをさがしますよね。ホンニャアにしても同じこと、長年培われた体験がさがしものにはモノをいうようです。

ホンニャアは体内に上等のセンサーがあるから、日に日に近づく灼熱の季節を、もう感じている。うすぐもりの空と湿った風にくるまれて、太陽はまだ休息しているのだ。

アイハラさんちのハチが、顔を天に向けて鼻をヒクヒクさせているけれど、あいつは犬だからまだ気づいてはいまい。そう彼は思う。猫の中にもにぶいのはいる。背中に座布団をのせたようなデザインのザブなんか、おデコハエにも気がつかないほど感覚が悪い。

でも、ホンニャアにはわかるのだ。ひかえめなようすを見せながら、けっこう大きな群れをつくって咲いているアジサイの花のかげで、もう夏は光りはじめていることを……。

光る床

つゆ明けはまだ先のことなのに、気温の上昇につれて、ホンニャアの体はアメがとけるようにだんだん伸びていく。彼の体の伸び縮みは温度計のようだ。そしてわが家の木の床板とすだいに仲良しになって、ダラリ、ベタリとはりついて過ごす時間がふえていく。

床張りをほどこしたものを、このごろではフローリング (flooring) としゃれた呼びかたをするらしいが、正方形をつなぎ合わせた木目の床は、夏の午後なら、猫でなくても寝そべってみたいくなる。

木の性質のふしぎさは、夏はひんやりとした感触でやすらぎを与えてくれるのに、冬は冬で独特のあたたかさをただよわせることだ。どちらにしても、きれいに磨きあげておくことで、いっそう心地よさが増してくるのは、おそうじ担当者だけの満足だろうか。

毛皮をまとったホンニャアの夏はさぞたいへんだろうが、天然のクッションのような体は床にべったりとはりつくことができて、なんともうらやましい。人間ではそうはいかないし、材質のとりあわせも毛皮と床の対比にはかなわない。

湿度の高いこの午後、ホンニャアは庭に近いリビングの床に、戸外をながめるポーズでよこたわっている。食卓の脚もとごしの、白く照り映える床に逆光のホンニャアがいて、静けさがあつた。

しかし、彼のセンサーはけっして休むことはない。私がめくるかすかな紙の音や冷蔵庫のうなり声に、耳が小さく動き、しっぽが緊張する。まるで後頭部にも目があるようだ。

ふと、いたずら心がおこって、私はホンニャアにさそいをかけてみる。

「ホンニャア、コロコロンは？」

庭を向いていたホンニャアは反射的に上半身をひねって起こし、あたりの床をキョロキョロとみまわした。

「コロコロン」とは、ビー玉が床をころがる音の擬音なのだ。トオルが小学生のころ、床にビー玉をころがしてはホンニャアをじゃれさせて遊んだ。ホンニャア自身もビー玉との追いかっけは好きなようだったが、私たちがあまり楽しそうなので、いっしょうけんめいサービスしているふうもあった。

「コロコロン」の言葉は、ビー玉をころがすたびに、「ニャアちゃん、コロコロン！」とくりかえしていたのを、いつのまにかおぼえたのだ。

もう何年も前の遊びを、ホンニャアがおぼえているだろうか？と試すつもりもあったのだが、「コロコロン」の情景は一瞬に彼のCPUからとびだしてきた。どこかの方向

から光りながら走ってくるガラスの玉をさがして、ホンニャアの目もビー玉のようになった。耳には、木の床をころがるあの「コロコロン」の音が聞こえはじめていたにちがいない。

記憶のすき間

「コロコロン」の遊びを思いだしてしまったホンニャアは、のんびりと休むのはやめて、さがしものをはじめた。果物やワインの乗った赤いワゴンテーブルの下を、まずのぞいている。そうだ、以前はここにビー玉の入った小さな籐 (とう) のカゴがあった。よくおぼえているものだ。あれをみつけたら、ビー玉遊びができると思ったのだろう。

子猫のころ、ポリエチレンの包装ひもでこしらえたボンボンが大好きで、遊びたくなると自分でくわえてきて、私たちの前にボンと投げだした。クールでわがままな彼だけれど、遊び以上に、私たちとの交流を望んでいるようすがしばしば感じられて、驚くことも多かった。

「コロコロンをさがしてるの？」

私はホンニャアに聞いてみた。

「ウン、どこにあるの？」という目で、ホンニャアは私を見上げる。

「どこかなあ」と、私はオーディオのラックのあたりをさがしてみせる。ホンニャアもイソイソと、私の横でいっしょにのぞきこむ。

夕飯をやるたび、「ゴハンゴハン」と語りかけていたら、とうとう猫が「ゴハンゴハン」と言うようになった話を聞いたばかりだったので、いまにそんなことが起こるかもしれない期待をかけて、ホンニャアと「会話」してみた。心がひとつになって、お互い意味することを伝えあえれば、それはき

っと会話と同じなのだ。

ところで、ビー玉はトオルの部屋にしまっているのだから、ホンニャアには申しわけないことになった。

「コロコロン、あるかな？」と、私はころがっているビー玉をさがすふりをして、カーテンの陰をのぞく。ホンニャアもついてきていっしょにカーテンの下に首を入れている。どうやってこの場をごまかさうかなと思っていると、ホンニャアはこんどは食器戸棚と冷蔵庫のすき間に小さな腕をつっこんでかきよせている。

細いすき間に腕のつけ根まで差し込んでいっばいに伸ばし、つかえた顔を横向けて手の先に注意を集中しているようすがあまりにおかしい。

「あるわけないでしょ……」と思わず人間相手の調子で言いかけたとき、ホンニャアがこちら向きになって、同時にホコリまみれの丸いものがころがり出してきた。

「あらァ……」と拾いあげてみると、それはビー玉よりはあまりに小さな、オモチャのガンにつめる弾丸だった。それでも、とりあえずコロコロンの代替品をみつけ出したホンニャアに私は敬意をあらわした。

ホンニャアは自分の記憶と経験から、ビー玉のたくわえられている本拠地をたしかめてみたり、それがころがって隠れこみするところをいくつか推理してみた。頭の中でじっさいにビー玉をころがして、第一の候補になったのが、冷蔵庫と食器戸棚のすき間だったのだ。

🐾 さがし屋稼業

「サーチャー」という技術者が、このごろ注目を浴びはじめて、その資格を得ようとする人がどんどんふえているようだ。

正確には「データベース検索技術者」といって、国内外のあらゆるデータベースから、必要な情報を引き出す専門家だ。基本的には、パソコン通信による各データベースへのアクセスと、必要事項の検索をするのだが、実務としての能力はなかなかたやすいものではないようだ。

「情報化社会」といまで言われてきたものも、コンピュータの活用によって、この数年でますます過密になった。現在日本で利用できる商用データベースは、海外のものが1800あまり、国内は420ほどで、5年

間で4倍になったそうだ。

ある特定の「情報」を得たいと思ったとき、情報源が大きく豊富であるほど検索は複雑になり、そのための専門の知識を持った技術者が求められることになる。日本でも、そういった時代の要求から、代行検索業の会社がつぎつぎ誕生している。

そんなところで力を発揮しようという、躍進的ともいえる特殊技能のしごと、それが「サーチャー」だ。

サーチャーをめざす人のために、情報科学技術協会が昭和60年から毎年実施している、「データベース検索技術者認定試験」がある。この試験には1級と2級があって、まず2級をめざしてみんな勉強する。2級は「与えられた機器を使用して、なんとかひとりで適切な検索を行い得る能力をもつ人」(情報科学技術協会資料より)で、1級は「2級の延長上のより高度なランクであり、単に自分が適切な検索を行うことができるのみならず、初心者、2級合格者を指導、管理できる能力を持つ人」(同)だそうだ。

このサーチャーになりたい人というのが、前記の資料によると、5年前の受験者は223人、うち合格者140人、合格者のうち女性は45%。昨年度は受験者816人、合格者301人、同じく女性58%で、女性の比率が大きくなってきている。

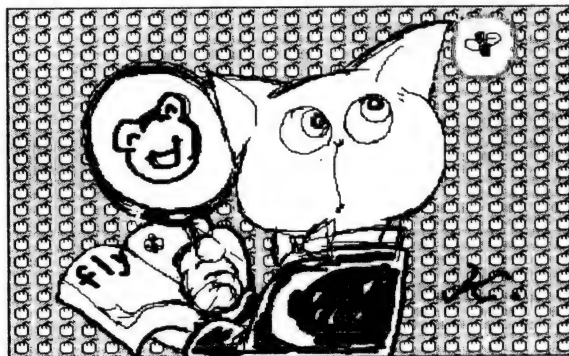
あるサーチャー講座の教室をのぞいたら、約35人の受講者のうち女性27人、男性8人で圧倒的に女性が多く、それもおおかたは20代だった。

単なるオフィスのオペレータとはちがう特殊な技術のいる職業として、なかなか魅力はあるものの、ただのカッコよさを求めているのでは少し甘いかなと感じる。

🐾 コロコロンの心

認定試験の問題もなかなかのむずかしさで、パソコン通信の知識はもとより、検索のための特殊なまり、専門用語、略語の解釈が山のようにある。その上、暗記しなければならない、あまりにたくさんのデータベースの種類、名称、特色。

空欄をうめる問題では、たとえば内容は



情報検索についての一般論であっても、同義語、類似語の微妙な判別がとてむずかしい。試験問題そのものが、検索者としての推理や分解や組み立ての能力をためしているようだ。

とはいえ、試験は正解の数が多ければいいのだ。若い人ほど暗記力はすぐれているし、「合格」はなんとかできるかもしれない。だが、そのあとの実務の世界は、マシンをあやつるだけの知識では、たやすく成り立たないらしい。

もちろんいちばんものをいうのは、各データベースの内容、特色を、自分の頭の中のデータベースにいかになくさん取り揃えているかということかもしれない。しかし目的は、依頼者の要求にいかになく適切に答えるかだ。

要求している人の目的や意図をじゅうぶんに理解する能力、その目的のために、という手順で検索をすすめていくかを組み立てる力。検索はかならずしもデータベースから始まるとは限らないようだ。ときには、それ以前に「要求された情報」に関する分野の、専門家の意見が必要になることもある。そういった知己を持っていることも、検索技術者の力の一部だという。

そして、広い範囲で知識が豊かで、経験も多いこと。なによりも、インスピレーションが鋭くはたらくこと。この直感がサーチャーの腕を左右し、海外データベースへのアクセス時間も最小限にしてくれることだろう。料金も重要な条件だ。

こうしてみると、サーチャーというしごととは、人とコンピュータ、それぞれの本質を深く理解できなければつとまらない、なかなかやりがいのある新職業だ。そして、そのスピリットは、なんといっても「コロコロン」をさがし出したホンニャアあのインスピレーションだ。

[第3話]

旅行あれこれ

TAKAHARA HIDEKI 高原 秀己

このところ、すっかりと海外旅行ブームは定着してしまい、もうブームなどと呼ぶのはふさわしくない。とくに年末年始や夏休みともなると、恒例行事といってもいいほどだ。日本人が海外の旅先で消費してくるお金は年間10億ドルだというからものすごい。

海外旅行にもいろいろな形式があるが、やはりバックツアーが一番の動員力を誇っているようだ。北海道や沖縄に行くのときとして変わらない金額で海外の人気旅行地に行ってしまうのだから、人気が出るはずだ。

いよいよ夏休み。

出不精のぼくも、せっかくの夏休みに何もしないのももったいないので、旅行代理店に足を運んで調べてみたのだが、バックツアーはさすがに安い。東南アジアやハワイ、グアムや東南アジアで10万円前後。15万円ちょっと出せば、アメリカ西海岸でもオーストラリアでも行けてしまう。

ところがいざ申し込んでみようとするとなかなか難しい。

「じゃあ、この12万8000円でオーストラリアっていうの、ありますか？」

「いっぱいですね。夏休みのピークの時期のは早めになくなってしまいますよ」

それで作戦を変更して夏休みをやや外してみることにしたのだが、それもなかなかうまくいかないようだ。

「8月末出発のシンガポール・マレーシア14万8000円っていうのはどうですか？」

「まだご予約がありませんね、何人様ですか？」

「ぼくだけです」

「それはどうですかねえ。おひとりですとツアーとして成立しませんので。他のお客様の申し込みを待って、ということになりますが、ご予約だけされますか？」

というわけで、旅行大作戦はひとまず延期することにして、旅行代理店から逃げ出してきた。

そもそも自分がカップルのひとりでないことが問題なのかもしれないのだが、それを気にしてはミジメになる。旅行代理店とバックツアーのシステムが、いや社会全体

の歯車が狂っていることにして一件落着としてしまったのだが、この分では夏休みは今年もたいしたことはできそうにない。

どうも男性がひとりぐらりと海外旅行をするっていうのは絵にならないようだ。そもそもがあまり、そういった不気味な客は想定されていないのだろう。

確かに雑誌でやっている旅行の特集企画にしても、ほとんど全部が女性向け。ある女性誌などは人気旅行地を毎号特集することに編集方針を変えてしまったほどだ。女性向け雑誌にはなくてはならないアイテムとなっている。

人気小説のトラベルミステリーなどにしても、たいてい事件を起こす客は女性かアベックと相場が決まっている。ひとり旅をする男性というのは刑事が探偵、あるいは出張しているビジネスマンと相場が決まっている。

かくいうぼくも、最近の旅行はスキーを除けば出張ばかり。

つい先日、九州を数日かけて回ってきた。久々に3日以上の上さで、旅行らしい旅行だった。

仕事とはいえ地方に行くと、緑と青の自然の景色を満喫できるので、なかなかの気晴らしになる。なんせ日頃は緑といえばゴルフ場くらいしか緑のない生活を続けているのだから。

今回はキーボードから離れた生活をしたかったので、昨年末のアメリカ旅行で移動端末機として大活躍してくれたラップトップパソコン(NECの4kgのマシン)はあえて持っていかなかった。

もっともヘッドホンステレオとゲームボーイはしっかりと持っていった。この2つは退屈な飛行機や列車の中では欠かすことができない小道具だ。

九州旅行での訪問先のひとつはA社の地方工場。そこに勤務する、ある課長さんと飲みに出かけた。

その課長さん、もともとは東京本社勤務の人なのだが、ここ数年は地方工場を転々としているそうなのだ。

アルコールが十分回ってきた頃、彼はと

ても面白い話をしはじめた。

「妻がいうんですよ。私はA社という企業社会の中で生活しているだけなんだから、東京本社であろうが、地方工場であろうがそれほどの違いはない。ところが自分はその地域の中で生活しなきゃいけないんだから、転勤があると影響をモロに受けてしまう。だから嫌だつてね」

これは盲点だった。

地方工場というのは、ロケーションこそたまたま地方にあるとはいえ、その企業の完全な一部分となって機能している。空間も工場という形で隔離されており、内部は企業社会の延長線上にある。

そこで働く人たちは県民とか町民という共通項でくくられているわけではなく、企業という名のパラレルワールドの住民なのだ。だから地方にいても、実際には地方で生活していることにはならない。

これは外資系企業のIBMとかTI(テキサス・インスツルメンツ)、インテルとかを考えると、さらにわかりやすい。

建物のデザインや内装からして、しっかりとそれぞれの企業カラーが打ち出されている。内部での生活様式ならぬビジネス様式も統一されている。

入り口を通り抜ければ、もう六本木の本社の中にいるのか、地方工場にいるのかすらはつきりしないほどだ。アメリカの本社ですら、違和感はない。

これからは企業が人々の生活に占めるウエイトがますます高まってきて、国や地域の差を吸収していくという説がある。

実際にこうした地方工場の機能を見ると、日本企業に限らず、国家とか自治体という縦割りの社会よりも強力な横割りの企業社会がジワジワと浸透してきているような気がする。

これについていける人についていけない人とは大きな違いが出てくるのだろう。

ちなみにその課長さん、さすがに3回目の転勤とあって、家族は東京近郊の家に戻ってしまい、哀れ単身赴任となっているそう。彼がいつ東京本社に戻るのか、まったく彼にもわからないそうだ。

BACK ISSUES

バックナンバー案内

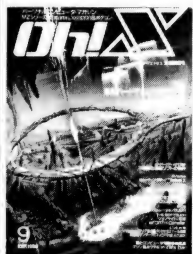
ここには1989年8月号から1990年7月号までをご紹介します。現在1989年7〜12、1990年1〜7月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については、176ページを参照してください。

1989



8月号

特集1 X1プログラミングガイドブック
PCGの基礎から奥義まで/超高速ラインルーチン 他
特集2 3Dグラフィックの深淵へ
スキャンラインZバッファ/3Dモデリング 他
【新連載】 (で)のショートプロバレー
X68000マシン語プログラミング/C調言語講座 PRO-68K
X-BASICプログラミング調理実習/D6GA・CGA講座
MZ-2500用グラフィックエディタ/Z80's Bar 他
全機種共通システム CP/M用ファイルコンバータ



9月号

特集 活用ハードディスク&プリンタ
各社ハードディスク接続総チェック/ハードディスク雑学
講座/COPYキーメニュー/ビデオプリンタ活用プログラム 他
THE SOFTOUCH ジェノサイド/琉球/mFORTH Compiler
●サイバースティックで遊ぶ 不思議な環境ソフトの世界
●X1/X1turbo用シューティングゲーム Defeat X
Z80's Bar/MZ-2500グラフィックエディタ 他
[X68000] X-BASIC/マシン語/C調言語講座/D6GA・CGA
全機種共通システム 生物進化シミュレーションBUGS



10月号

特集 ゲーム面白心理学
ソーサリアン・宇宙からの訪問者/ファンタジーゾーン
ねじ式/ガウディ・バルセロナの風/サバッシュ 他
●MZ-700用シューティングゲームSide Roll-F
●X1/X1turbo用カードゲームBonding
ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/D6GA・CGA
THE SOFTOUCH Z'sTRIPHONY DIGITAL CRAFT/James68K
全機種共通システム 小型インタプリタ言語TTI



11月号

特集 microComputer入門
初歩からのCPU物語/RISCプロセッサの設計と製作
X68000&X1で周辺LSIを使いこなそう
【連載】 ショートプロ/Z80's Bar/MZ-2500グラフィックエディタ
X68000マシン語/X-BASIC/C調言語講座/D6GA・CGA
●X68000用カードゲームばばぬき
LIVE in '89 メタルホーク/オブ・ラ・ディ、オブ・ラ・ダ
THE SOFTOUCH Stationery PRO-68K/リングマスター1
全機種共通システム TTI用パズルゲームPUSH BON!



12月号

特集 Cプログラミングへの招待
付録 C言語簡易リファレンス
【連載】 ショートプロバレー/Z80's Bar
X68000マシン語/X-BASIC/D6GA・CGA
●Oh! X2周年特別企画「素粒子の音が聞こえる」
●X1/turbo用アクションゲームACTIVE UNIT
LIVE in '89 天空の城ラピュタ/ギャラクシーフォース
THE SOFTOUCH 38万キロの虚空/た〜みのる2
全機種共通システム SLANG用リダイレクションライブラリ



1月号

特集1 オペレーティングスタイルの研究
特集2 Cプログラミング応用編
【連載】 ショートプロバレー/Z80's Bar
X68000マシン語/C調言語講座/D6GA・CGA
●X1/turbo用シミュレーションゲームSuper Battle
LIVE in '90 さよならを過ぎて/RIDEEN
THE SOFTOUCH レナム/メタルサイト
全機種共通システム WORM KUN/再掲載SLANG
特別付録 X68000 THE SOFTWARE CATALOGUE



2月号

特集 画像圧縮へのアプローチ
【連載】 ショートプロバレー/Z80's Bar/D6GA・CGA
X68000マシン語/C調言語講座/X-BASIC調理実習
●X68000用ゲームプログラムGon Gon
●MZ-700用紙芝居Eyelarth
LIVE in '90 オーダイン/魔女の宅急便
THE SOFTOUCH A-JAX/フラッピー2/夢幻戦士ヴァリスII
マジックバレット/Mu-1/CYBERNOTE PRO-68K
全機種共通システム 超小型コンパイラTTC++



3月号

特集 MUSICアドベンチャー
X68000用MIDIドライバ&音源エディタ
なんでも鳴らせるOPMD.X/MMLを楽譜データに
【連載】 ショートプロバレー/Z80's Bar/D6GA・CGA
C調言語講座/X-BASIC調理実習
●X1/turboシミュレーションCRISIS in Tokyo
LIVE in '90 パワードリフト/スキーム/どりのトロ
THE SOFTOUCH ナイトアームズ/斬/ダンジョンマスター
全機種共通システム 超多機能アセンブラOHM-Z80



4月号

特集 ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
【連載】 ショートプロバレー/Z80's Bar/D6GA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040、ついに登場
LIVE in '90 バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム ファジコンピュータシミュレータ-MY



5月号

特集 BASICプログラミング
第5回 言わせてくれなくちゃだワ
【連載】 ショートプロバレー/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90 TURBO OUTRUN
THE SOFTOUCH 天下統一/ポピュラス/Hyperword
全機種共通システム インタプリタ言語STACK



6月号

特集 創刊8周年記念PRO-68K(付録5"2HD)
Oh! Xアンケート結果大分析大会
【連載】 ショートプロバレー/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1 turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90 ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH 三国志II/FAR SIDE MOON/グラナダ
全機種共通システム X68000用S-OS"SWORD"他



7月号

特集 マシン語への第一歩
X68000SUPER-HD試用レポート
【連載】 ショートプロバレー/Z80's Bar/D6GA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1——ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90 夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH サークあーくしゅ/ダウンタウン熱血物語
全機種共通システム リロケータブルアセンブラWZD

1990

NEW PRODUCTS

スーパーアウトラインフォント内蔵

WD-A320/340

シャープ



WD-A340

シャープは「見やすい大型液晶画面」、「活字に迫る高品位印刷」、「思いどおりのレイアウト」、「正しいことばづかい」などを追求したラップトップ型ワープロ「WD-A320」および「WD-A340」を発売した。

「WD-A320/340」は新開発の専用LSIにより名刺用の小さな文字から拡大文字まで美しくなめらかに印字する、「書院スーパーアウトラインフォント」を内蔵している。曲線データで文字を形成しているため、直線（ベクトル）データによるアウトラインフォントに比べ品位を向上している。4.5～288ポイントまで合計67種類のマルチポイント文字（欧文時はマルチポイント23種類）を自由に設定することで、多彩な大きさの文字を利用できる。また、それに加えて64ドット・400DPIの高精細プリンタを搭載していることで、美しい印字が可能となっている。

さらに、パーソナルDTP機能、手紙文の作成に便利な「直子の代筆（書院版）」、15万例のAI-V3辞書、電子手帳とのデータの共有ができる電子手帳機能などの機能も装備している。

「WD-A340」ではこれに加えてハイコン

トラスト白黒液晶画面、類語辞書、文体統一機能などの文書校正支援機能、MS-DOSコンバータ、通信ソフトなどを搭載している。価格はそれぞれ178,000円と198,000円（どちらも税別）。

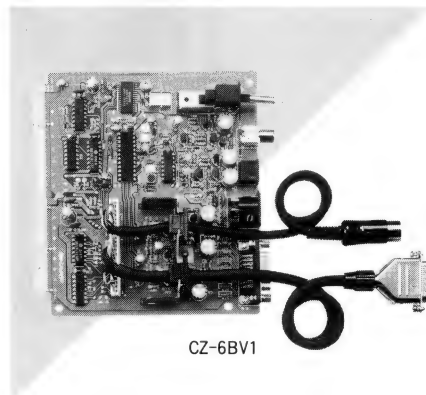
〈問い合わせ先〉

シャープ(株) ☎06(621)1221,03(260)1161

X68000用ビデオボード

CZ-6BV1

シャープ



CZ-6BV1

シャープはX68000用の周辺機器としてビデオボード「CZ-6BV1」を発売した。このボードをX68000の拡張I/Oスロット（2スロット分を使用）に装着することにより、コンピュータ映像をビデオ信号として取り出すことができるようになる。たとえば、X68000上で作ったグラフィックやアニメーションあるいはゲーム画面などを手軽にVTRに録画することができる。さらに、ビデオ入力端子のついている液晶ビジョンや大型テレビにX68000を接続して、迫力ある大画面でゲームなどを楽しむこともできるようになる。特徴は以下のとおり。

- ・ NTSCエンコーダ、同期信号発生回路とも1チップ化

- ・ 入出力端子は以下のものを装備

- アナログRGB×2

- テレビコントロール×2

- S映像出力×1

- コンポジットビデオ出力×1

- ・ 高解像度モード時のビデオ出力を自動的に停止することができる

価格は21,000円（税別）。

〈問い合わせ先〉

シャープ(株) ☎06(621)1221,03(260)1161

X68000とMacをリンク

Mac版「XIN/XOUT」

電機本舗

電機本舗はRS-232Cを介してデータ転送をするシステム、「XIN/XOUT」のMacintosh版を発売した。これはRS-232C/422通信ポートを利用して、Macintosh Plus, SE, SE/30, IIとMS-DOSマシン/X68000の間でのファイル転送を可能にするものである。バイナリファイルの転送も可能で（エラーチェックは独自のものを採用）、ファイルの一括指定一括転送もサポートしている。転送に際しては、転送先のファイル形式に自動変換、OSの相違を完全吸収し漢字を含んだファイルも正確に転送する。英語、日本語環境およびマルチファインダ上にて動作する。

パッケージにはRS-232Cケーブルと、ファイル転送プログラムのMac版とMS-DOS（/X68000/PC-DOS）版のフロッピーディスク2枚が入っている。価格は12,800円（税別）。

〈問い合わせ先〉

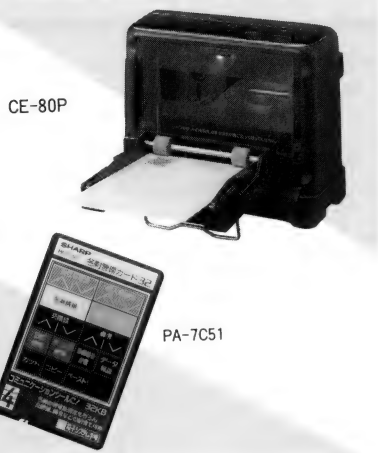
（有）電機本舗 ☎03(447)1773, BBS

03(447)2564 1200bps

XIN/XOUT



電子手帳用プリンタ&名刺管理カード
CE-80P, PA-7C50/7C51
シャープ



シャープは既存の電子手帳すべてに接続可能なプリンタ「CE-80P」を発売した。さらに、面倒な名刺の整理に便利な名刺管理カード「PA-7C50/51」を7月25日に発売する。

電子手帳用プリンタ「CE-80P」ははがきやラベルへの宛名印字はもちろん、リフィルへの住所録印字もできる。別売のはがきフィーダを装置すれば、連続20枚までのはがき裏面の連続印字が可能。年賀状などで使うあいさつの慣用句73種類を内蔵しており、また、オプションの毛筆体カートリッジ「CE-61M」により美しい毛筆体での印字が可能になるので年賀状などが簡単に作成できる。リボンカセットは黒、赤、青、茶、金、銀が用意されていて（茶は8月発売予定）、6色印字が可能。価格は45,000円（税別）。

名刺管理カード「PA-7C50/51」は名刺情報はもちろん、いつ、どんな用件で会ったのかを記憶できる交際録、趣味や嗜好を記憶できる備考、年賀状やお歳暮などの状況をチェックできるチェックリストなどの記憶が可能。名刺情報は名前4文字、電話番号12桁、FAX番号12桁、会社名8文字、所属5文字、役職2文字、郵便番号3桁、住所20文字の場合で約350人分（PA-7C50の場合は約160人分）が記憶できる。機能としては郵便番号辞書、日付検索やチェック検索などの多彩な検索機能、宛名印字機能を搭載。さらに本体メモリをバックアップできるRAMファイルとしての使用も可能

となっている。価格は「PA-7C50」が13,000円、「PA-7C51」が16,000円。

〈問い合わせ先〉

シャープ(株) ☎06(621)1221,03(260)1161

32ビット浮動小数点DSP
DSP96002
モトローラ

モトローラは24ビット固定小数点デジタルシグナルプロセッサDSP56000ファミリの上位機種として、32ビット浮動小数点DSP96002を開発した。

- ・動作周波数：27MHz,33MHz
- ・命令サイクル：74nsec,60nsec
- ・IEEE754データフォーマットに準拠
- ・43×43ビット→96ビット浮動小数点演算
- ・32×32ビット→64ビット整数演算
- ・12Gワードのメモリ空間
- ・1KワードのオンチップデータRAM
- ・1KワードのオンチップデータROM（サイイン、コサインテーブル）
- ・512ワードのオンチッププログラムRAM

- ・2チャンネルDMAC
- ・32ビットバレルシフタ
- ・223ピンセラミックPGAパッケージ
- ・割り算と平方根用に高速な命令（6命令サイクルと9命令サイクル）を用意

DSP96002の2つの外部メモリ拡張ポート（ポートAおよびポートB）はユーザープログラミングによって、外部メモリのアクセスポートあるいはホストプロセッサとの接続ポートとして使用できる。さらに、DSP96002の各ポートにはマルチプロセッサ構成をサポートする信号線も用意されているので、複数のDSP96002でマルチプロセッサを構成し高性能な演算処理を実現することもできる。

以上のような特長により、DSP96002は従来のDSPでは処理が困難であった画像処理、浮動小数点演算アクセラレータ、医用機器、周波数解析処理などに応用が可能である。

〈問い合わせ先〉

モトローラ(株) ☎0120-068030

I N F O R M A T I O N 番外編

「X68000グッズショップ in Akihabara」

ミナミ電気株式会社 本館5階

X68000グッズが買いたいと思っても、いままでは常備店がなかったの、イベントに行き行って買うなどしか方法がありませんでした。しかし、このたびミナミ電気本館5階のパソコンフロアにX68000グッズショップ in Akihabaraが開設されることになり、いつでもX68000グッズを手に入れることができるようになります。

そこで、それを記念してひょっとしたらあまり知られていないかもしれないグッズの数々を紹介してみたいと思います。

★X68000牛革ベルト

標準価格6,300円（税別）
バックルには光輝く“X”のロゴが……

★X68000キーホルダー

標準価格1,300円（税別）
X68000の電源スイッチにも鍵があればよかったのに

★X68000ネクタイピン

標準価格3,000円（税別）
ネクタイをする人にはいいかも

★X68000電飾POP

標準価格9,500円（税別）

暗い所で見ると本当にきれい

★X68000クリスタルボルシェ

標準価格8,000円（税別）

ガラスでできたボルシェ911

★X68000ジッポ・ライター

標準価格4,800円（税別）

あのツタンカーメンの仮面が……

さらに、

★X68000ゴルフボール

標準価格1,900円（税別）

★X68000傘

標準価格4,200円（税別）

★X68000スポーツタオル

標準価格3,300円（税別）

と、「こんなものまで？」と思うような変わった(?)商品が、ほかにまだまだいろいろあります。興味のある方はお店でご覧になるとよいでしょう。

☆万世橋交差点際 第一家電機



牛革ベルト



キーホルダー/タイピン



電飾POP



ジッポ・ライター



ゴルフボール

FILES Oh!

このインデックスは、タイトル、注記——
筆者名、誌名、月号、ページで構成されて
います。毎日暑い日が続きますね。夏バテ
や寝冷えに気をつけて、楽しく有意義な夏
休みを過ごしてください。

一般

▶特集シムアース

シムシティの登場によって示されたパソコンシミュレーションの楽しさ。今度はもっとグローバルに地球環境のシミュレーションをやってみよう。そこで発表されたのが「シムアース」。その概念や裏話などを解説。シムアースを考える座談会にはミュージシャンの細野晴臣、戸田誠司、日本自然保護協会の横山隆一らが参加している。——編集部、LOGIN、12号、116-127pp.

▶ネットワーク・ホリック 第22回

新聞の申し込みまでできちゃうぞ。大手ネットのショッピングサービスを紹介。PDSはPC-9801のZMODEM転送プログラム「ZM.EXE」、X68000のシューティングゲーム「MEMORY BROKEN.X」。全国BBS探訪記は秋葉原にあるPENCIL-NET。——編集部、LOGIN、12号、202-203pp.

▶ハードラボラトリー

MIDIについて解説。X68000の純正MIDIボードCZ-6BMIやMusicstudio PRO-68Kも紹介。——編集部、POPCOM、7月号、106-108pp.

▶X68000のウイルス騒動の真相

先頃新聞を騒がせたX68000用市販ソフトへのウイルス混入事件についてウイルス騒動の当事者が内情を語る。日コン連では昨年11月に各マスコミへ今回のウイルスのソースリストを送っていたという。——日コン連理事長山本隆雄、The BASIC、7月号、176-177pp.

▶2大ショウに見る最新パソコンの現状

ビジネスショウ・マイコンショウに展示された各社の新製品をレポートし、今年のトレンドを探る。——編集部、マイコン、7月号、135-144pp.

▶コンピュータ・ウイルスを考える

ウイルスについて正しい理解をするために、ウイルスの種類や事例、対策について述べる。——コンピュータ・ウイルス研究会、マイコン、7月号、164-165pp.

▶楽器が弾けなくても、声で楽器が演奏できる

マイクロコンピュータショウに出展されていた、ボイスインプットを紹介。マイクに入力された音程を解析してMIDI楽器を鳴らすことができる。——FORESIGHT企画部・藤本健、マイコン、7月号、239-240pp.

▶ビジネスマンの情報管理術

著者のヨーロッパ旅行記第3弾。ポルトガル、オランダ、イギリスなどで7カ国語翻訳カードと通貨換算機能が活躍する。——塚田洋一、マイコン、7月号、310-312pp.

▶やまさんのアルゴリズム・ブック

MS-DOSなどで頻繁に使われるワイルドカード機能のアルゴリズムを考える。——やまさん、マイコン、7月号、321-325pp.

▶実践ハード入門

梅雨にあわせて、湿度センサを使った簡易湿度計を作る。——石川至知、マイコン、7月号、334-336pp.

▶レーザーディスクで広がるマルチメディアの世界

レーザーディスクの生み出すハイパーメディアの世界について述べ、またマッキントッシュでのハイパーメディアの現状を報告する。——田島恵介・長谷川昌夫、マイコン、7月号、346-354pp.

▶NEW MACHINES '90

NEC、エプソンなどの新機種と共に、AX仕様のAll in Note、X68000SUPER-HDを取り上げ、概要を紹介する。——編集部、ASCII、7月号、258-280pp.

▶AtariSTの魅惑の世界

68000使用のホビーパソコン、米Atari社のSTシリーズの魅力に迫る。今月はラインナップ、ハードウェア、PDSやゲーム事情などについて。——小沢靖・池田賢司・判治聡、ASCII、7月号、313-320pp.

▶MEDIA BREAK

北九州市八幡にオープンしたスペースワールドの宇宙飛行士訓練プログラム「スペースキャンプ」を紹介。——浦山明俊・佐藤守弘、ASCII、7月号、409-411pp.

MZシリーズ

MZ-1500 (MZ-5Z001 BASIC)

▶1582

カプコンのシューティングじゃないよ。戦国アクションゲーム。——大石豊、マイコンBASIC Magazine、7月号、126-128pp.

MZ-2500 (BASIC-M25)

▶BLOCK BROKEN

ブロックと入れ替わる難解パズルゲーム。——Tak KuN、マイコンBASIC Magazine、7月号、129-130pp.

▶Multi Window

BASICのウィンドウサブルーチン。——佐藤拓也、マイコンBASIC Magazine、7月号、179-180pp.

X1/turbo/Z

X1シリーズ

▶最新ゲーム徹底解剖!!

新着ゲーム「スライマ」の基礎攻略法を紹介。——編集部、LOGIN、11号、226-227pp.

▶攻略おすめゲーム

ウィザードリィVの地下3階までを攻略。——編集部、テクノポリス、7月号、50-53pp.

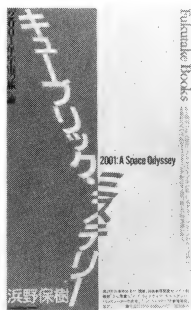
▶桃四郎

好評の桃シリーズ、今回は桃太郎4人目の兄弟の話。お供をやといて鬼をたおすアクションゲーム。ジョイスティック

参考文献

I/O 工学社
ASCII アスキー
コンプティーク 角川書店
The BASIC 技術評論社
テクノポリス 徳間書店
POPCOM 小学館
マイコン 電波新聞社
マイコンBASIC Magazine 電波新聞社
LOGIN アスキー

新刊書案内



この人の著書(「ハイパーメディア・ギャラクシー」など)を読むと、実に「2001年宇宙の旅」に関する話が多い。趣味が高じてか今度は「2001年宇宙の旅」を中心においた映像論の本を書いた。本書は2つの点で実に面白い。ひとつは、そこらの映画評論家を書く映画評より資料も視点もしっかりしていること。もうひとつは、どうして著者はコンピュータはメディアを目指すべきだと考えるのか。メディアとなったコンピュータに何を期待するのかははっきりとわかることだ。

オーソン・ウェルズ、小津安二郎、そしてキューブリックの3人の映画監督の共通点。彼らは何

と戦い、何を表現しようとしたのかということ。「ジョージ・ルーカスやスティーブン・スピルバーグは、最新の特撮技術を総動員して、過去のイメージを増幅しているだけだ」ということ。HALはなぜ殺さねばならなかったのかということ(2010年で示されたような安易な答えではない)。「2001年宇宙の旅」はメディア論だということ。著者はメディアとしてコンピュータを使うことによって、個人の表現を復権させたいのである。(K)キューブリック・ミステリー 浜野保樹著 福武書店

☎03(230)2131 新書判 204ページ 1,130円

ック専用。——ズオ、マイコンBASIC Magazine, 7月号, 158-160pp.

▶LEADER LEADER

シルクハットをかぶったハット君にパンを食べさせてゴールに向かう。風船で道をつくってハット君を誘導する。風船パズルゲーム。——吉川章、マイコンBASIC Magazine, 7月号, 161-164pp.

▶性格判断

学園祭の定番、性格判断プログラム。多少判定の文章が貧しいという声もなくはないが……。——編集部、マイコン, 7月号, 212-216pp.

X1+FM音源ボード(要NEW FM音源ドライバ)

▶ミスティ・ブルー

エニックスのアドベンチャーゲームのミュージックプログラム。——KENJI、マイコンBASIC Magazine, 7月号, 192-194pp.

X1 turboシリーズ

▶NEW SOFT

セレクトッドソーサリアン4のシナリオの解説。——編集部, LOGIN, 12号, 12-13pp.

▶攻略おすすめゲーム

世界の海を股にかけるゲーム、「大航海時代」の最も重要な要素、交易について攻略。——編集部, テクノポリス, 7月号, 46-49pp.

▶月に帰りたいヒトデちゃん

降ってくる星を足場にして月まで帰る。スクロールアクションゲーム。——HARU、マイコンBASIC Magazine, 7月号, 164-165pp.

X68000

▶NEW SOFT

7月発売予定の「ウルティマV」と「闇の血族」、そのほか発売中の「バズニック」「天下統一」「ダウタウン熱血物語」を紹介。——編集部, LOGIN, 11号, 12-25pp.

▶X68000新聞

戦国ゲーム特集。「天下統一」をはじめ「信長の野望・全国版/戦国群雄伝」「斬(ZAN)」を紹介。そのほか「POOL BAR」「闇の血族」「ダウタウン熱血物語」「ガンシップ」を紹介。——編集部, LOGIN, 11号, 162-167pp.

▶最新ゲーム徹底解剖!!

新着アクションゲーム「グラナダ」の攻略・その2。ステージ4からステージ6までを、マップを載せて紹介。アクションパズルゲーム「スライミヤ」も紹介。——編集部, LOGIN, 11号, 196-199・226-227pp.

▶Software Review

ポピュラスを真面目に考えてみる! ほかのゲームとはちょっと違うポピュラスの面白さは? ——川村B, LOGIN, 11号, 230-231pp.

▶NEW SOFT

8月発売予定のシミュレーションゲーム「JOSHUA」、7月発売予定の「POOL BAR」を紹介。——編集部, LOGIN, 12号, 19・22p.

▶X68000新聞

新着ゲームの紹介。「ラグーン」「維新の嵐」「ルーンワース」。そのほかジェノサイドのCDレコーディング風景やThe File Professorの解説。——編集部, LOGIN, 12号, 130-135pp.

▶先取りおすすめゲーム

7月中旬発売予定の「ラグーン」を紹介。——編集部, テクノポリス, 7月号, 14-15pp.

▶GAMING WORLD

好評のくにおくんシリーズ「ダウタウン熱血物語」、アクションパズルゲーム「バズニック」「スライミヤ」「タッグ・オブ・ウォー」、発売予定の「ユニオン」「レインフォース」「RYU〜哭きの竜より〜」を紹介。——編集部, テクノポリス, 7月号, 18-30pp.

▶攻略おすすめゲーム

第二次大戦のフランス戦をあつかった陸戦シミュレーションゲーム「機甲師団」を徹底攻略。——編集部, テクノポリス, 7月号, 56-57pp.

▶レモンちっくWORLD

発売予定の美少女RPG「ランス2〜叛逆の少女たち〜」、麻雀ゲーム「びんびん麻雀ピーチエンゼル」、カードゲーム「DOKI DOKI Card League」を紹介。——編集部, テクノポリス, 7月号, 72-79pp.

▶SLGの夏が来た!!

シミュレーションゲーム特集。ポピュラスの紹介やその原作者ピーター氏からのありがたいお告げなど。——編集部, POPCOM, 7月号, 62-63pp.

▶WE ARE THE X68000 WORLD IN HOKKAIDO

新着ゲーム「ラグーン」「POOL BAR」「Vessel」「サーク」「ルーンワース」「レインフォース」「ユニオン」などとスプライトツール「びくせる君」を紹介。——編集部, POPCOM, 7月号, 68-72pp.

▶ゲームがオレを呼んでいる!

くにおくんシリーズ「ダウタウン熱血物語」と発売予定のゲーム「ウルティマV」の攻略法を解説。——編集部, POPCOM, 7月号, 82-90pp.

▶パズルDEバトル

新着パズルゲーム「バズニック」を紹介している。——さすらいのバズラ, POPCOM, 7月号, 92-93pp.

▶ミュージックパビリオン

映画「香港パラダイス」の主題歌「無敵のビーナス」(GO-BANG'S)のミュージックプログラム。——編集部, POPCOM, 7月号, 176-179pp.

▶キミのX68000を護れ!

コンピュータウィルスの基礎知識ほか, X68000のIPL,

SRAM常駐型ウイルスに対して有効なワクチンソフトを誌上公開。——GORRY, マイコンBASIC Magazine, 7月号, 67-73pp.

▶誌上公開質問状

X-BASICの画像フォーマット「GL3」の解説や、カラーイメージユニット「CZ-6VT1」の機能紹介。そのほかCommunication PRO-68KでATモデムは使えるか? などの質問に答えている。——多田太郎, マイコンBASIC Magazine, 7月号, 90p.

▶わかった!

画面に隠れたアルファベットを当てる。マウス専用、文字さがしゲーム。——小野正明, マイコンBASIC Magazine, 7月号, 166-167pp.

▶PYRAMID BREAK

ピラミッド型につまめた5種類のブロックを落とさずにとっていく。山くずしゲーム。——萬道賢治, マイコンBASIC Magazine, 7月号, 168-170pp.

▶リレーレビュー

ウルフ・チームの「グラナダ」について、4人のライターの意見を聞く。——編集部, マイコン, 7月号, 194-195pp.

▶スクリーンエディタEDX

Human68kとOS-9/X68000上で共通の操作環境を提供するスクリーンエディタ。いわばEDXの機能強化版である。——村田誠, ASCII, 7月号, 335-338pp.

▶AV STRASSE

PDSのグラフィックエディタ, MFGEDを紹介。高機能ではないが瞬時に立ち上がる小回りの良さが身上。——仲田津弘, ASCII, 7月号, 353-356pp.

▶NEWBAT.X

以前発表されたBATKEY.Xのバージョンアップ版。バックファイアの機能を拡張してくれる。——牛島健雄, I/O, 7月号, 198-202pp.

▶迷路エディタ

最大511×511のマス目にマウスで絵を描くと、それを正解として迷路を作ってくれるというもの。——カバウシ2世, I/O, 7月号, 189-197pp.

ポケコン

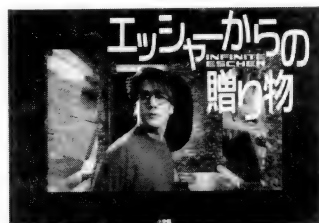
PC-E500

▶TURBO RUN

ドライブングゲーム。——森高周作, マイコンBASIC Magazine, 7月号, 175p.

▶DRAGON BUSTERD

ドラゴンバスターことクローブス进行操作してドラゴンをやっつける。アクションゲーム。——広鹿太一, マイコンBASIC Magazine, 7月号, 176-177pp.

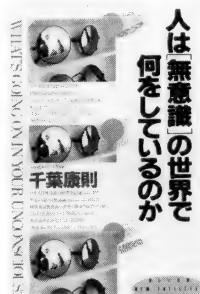


エッシャーからの贈り物

エッシャーの描いた数々の作品を、CGで表現した。同じ内容のビデオも発売されており、そちらのほうがメインのようだ。作品の質としては今ひとつの感があるが、ビデオで見るとまた違った味わいだろう。エッシャーの騙し絵をCGにしようという発想はなかなかよい。(K)

野崎昭弘著 小学館

☎03(230)5442 B5判 47ページ
1,680円



人は「無意識」の世界で何をしているか

無意識の世界。カッコよくいうと、サブミナルとか潜在意識とかとなる。本能や反射など、とにかく、人間のほとんどの活動は意識に現れないところで行われている。自分は意志に基づいてのみ行動していると思っている人、これを読んで謙虚になりなさい。PHPくさいところがわずかにあるが、丁寧な語り口で脳と無意識と行動の話を紹介している。専門的な内容はほとんどない。わからないことはわからないとしているのも善良。(K)

千葉康則著 PHP研究所

☎03(239)6221 B6判203ページ 1,000円



X68000のアセンブラで乱数発生
 生のプログラムを組もうと思う
 のですが、乱数発生原理がわ
 からないで困っています。乱数発生原理（乱
 数は1ロングワードの整数）はどうなっ
 ているのでしょうか？ 徳島県 森上 晶仁



一般に乱数は線形合同法と呼ば
 れる方法で作られています。こ
 れはある式に値を代入して計算
 によって乱数を生成する方法で、詳しい説
 明が1988年8月号に紹介されていますから
 興味のある方はそちらをどうぞ。

ところで、X68000には乱数を生成するた
 めのファンクションコールが用意されてい
 ますから、それを利用することにして使い
 方を説明しましょう。

まず、このファンクションコールはFLO
 ATn.Xを組み込むことによって使えるよ
 うになるものです。乱数発生部のコール番
 号は\$FE0Eとなっていますのでアセンブ
 ラで書くなら、

```
dc.w $FE0E
```

もしくは、FEFUNC.Hをインクルードし
 て、

```
FPACK RAND
```

(戻り値はd0.w)

という具合に使うことになります。

また、乱数系列の初期化には、

```
dc.w $FE0D
```

```
FPACK SRAND
```

(引数はd0.w)

とします。内容はBASICのRAND(),
 SRAND()と変わらないと思います (たぶ
 ん)。

ここで得ることのできる乱数の値の範囲
 は、0から32767と森上さんの希望とは違
 うものですが、実際には32ビットの乱数を必
 要とされることは稀だと思いますし、もし
 必要なときはこの方法で得た乱数にビット
 シフトなどの加工をしてから、さらに乱数
 を加えとか、工夫次第でどうにでもなる
 でしょう。



編集室の皆様こんにちは。僕は
2年たってもろくにプログラム
の組めない大バカ野郎です。6

月号の付録のディスクはとてもよかったで
 す。大事に使わせてもらっています。僕は
 前からCGをやってみたいと思っていまし
 た。だからANGELが動くのを楽しみにし
 ていたのです。

いざ解凍してみてもコマンドモードで
 “ANGEL”と入力してみると、「主記憶が
 足りません」と出てきました。ASK68Kをは
 ずしてみなさいと書いてあったので、自分
 なりにはずしてみましたが同じメッセージ
 しかでてきません。もう一度ASK68Kをは
 ずすところからできるだけ詳しく書いてく
 ださい。機種はX68000ACE, Human68k
 Ver.1.01, メインメモリは1Mバイトです。

愛知県 藤田 聡



同じ内容の質問がほかにも何通
 か送られてきましたが、藤田さ
 んのハガキが一番最初に送られ
 てきました（往復ハガキは使わないでくだ
 さいね）。とにかくX68000というマシンは
 メモリを大量に必要とするマシンです。標
 準で1Mバイトしか積んでいないマシンを
 使っている方は、BASICから子プロセスを
 実行することもままならないでしょう。

普通に考えれば、メモリを増やすにはパ
 ソコンショップにいった増設メモリを買っ
 てこなくてはいいませんが、とりあえず使
 うことのないデバイスドライバを組み込ま
 ないようにしてメモリの空き容量を増やす
 ことも可能です。質問電話によると藤田さ
 んと同様のケースではほとんどがビジュア
 ルシェルの起動したためのメモリ不足で
 した。このあたりの話は先月号でも触れら
 れていましたが、もう少し詳しく話しまし
 ょう。

Human68kは起動したドライブに存在
 するCONFIG.SYSの内容に従ってデバイ
 スドライバの組み込みを行います。つまり
 ASK68Kなどのデバイスドライバを組み
 込まないということは、CONFIG.SYSの内
 容を変更することにほかなりません。それ
 にはエディタ、ワープロ、またはCUSTOM.
 Xのどれかを使うことになりますが、こ
 こではエディタを使って変更するとしまし
 ょう。まず、

```
ED A: ¥CONFIG.SYS
```

としてCOFNIG.SYSをエディタに読み込
 みます。この場合はED.Xがパスの通ってい
 るディレクトリにあり、CONFIG.SYSがド
 ライブAのルートディレクトリ上にあるも
 のと考えています。画面のどこかに、

```
DEVICE=¥SYS¥ASK68K.SYS...
```

といった行があるはずですから、それを

```
*DEVICE=¥SYS¥ASK68K.SYS...
```

と先頭に*を挿入します（*をつけると注
 釈行扱いとなる）。こうしてからESC・Eで
 ファイルをセーブしてエディタを終了させ
 ます。これでASK68Kを組み込まないシス
 テムの完成です（注：リセットして再起動
 しなくてははいけません）。

ほかにも登録したくないデバイスドライ
 バがあったら、同様の変更をすることで組
 み込まないようにすることができます。プ
 リンタドライバやPCMドライバもと
 りあ
 えずいらないでしょうし、間違ってもRAM
 ディスクを設定してはいけません。

また、Human68k Ver.2.0などには、

```
OPMDRV.X
```

```
HISTORY.X
```

```
FLOATn.X
```

```
IOCS.X
```

など、実行可能ファイルのくせにデバイス
 ドライバとして登録できるものがあります
 （このうち、必ず設定しなければならない
 のはFLOATn.Xのみです）。FM音源を使
 うならOPMDRV.Xをデバイスドライバと
 して登録するために、

```
DEVICE=OPMDRV.X
```

と書くことになっていますが、そうしなく
 ともコマンドモードから、

```
A: ¥SYS¥OPMDRV
```

とすれば、FM音源を使うことができますし、

```
A: ¥SYS¥OPMDRV OFF
```

とすれば、いつでもFM音源を使わないよ
 うにすることができます（使えなくなるだ
 けで空きメモリが増えるわけではない）。

ですからFM音源を使うことが減多にな
 いのなら、OPMDRV.Xを組み込まないよ
 うにしたほうがいいでしょう（標準1Mバ
 イトの方は特に）。OPMDRV.Xを使用する

ソフトを起動したときは、エラー(\$FE0D)が発生しますから、そしたらOPMDRVとコマンドモードから入力すればいいのです。こうしておけば、OPMDRV.Xを使わない場合は通常87000バイト、コマンドモードから登録した場合も、わずかで3000バイトほど空き容量が多くなります。また実行速度も割り込みが発生しない分だけ、いくらか上がります。

またIOCS.Xを組み込んでいる人もメモリが狭いと感じるようだったらはずしておくことをすすめておきます。スクロールの高速化などあれば便利ですが、なくても動くんだから我慢しましょう。また、FILESやBUFFERSの最初の数字も小さくすると多少はメモリ消費が抑えられます。ディスクアクセスが遅くなったり、同時に扱うファイル数に制限が出ますが「背に腹は代えられぬ」ってやつですね。

もちろん、このような操作も、ビジュアルシェルスで起動すると台無しです。真っ先にコマンドシェルスで起動するシステムディスクを作ってください。方法は各機種種の取扱説明書第3部「より高度な使い方」の3章「デスクトップを使わない操作」の4項「起動時にコマンドモードに入るには」を参照してください。



パソコンの画面をビデオに録ろうと思い、X68000のカラーイメージユニットを買ったのですが、市販のソフトウェアをビデオに録るときに、コンピュータの画面モードをスーパーインポーズすると黒が透けてテレビ番組が映ってしまいます。VCUTを実行しようとしても市販ソフトなので無理です。どうしようもありません。どうにかテレビ画面をカットする方法はないでしょうか。

静岡県 石井 孝



スーパーインポーズの状態でないでビデオ録画できないという制約がなければなんでもないのであるのですが、どんなに考えてもスーパーインポーズさせないで録画できないのは仕様上、変更することは無理だと判断できません。

問題点はスーパーインポーズにあるのではなく、黒色が透明扱いされてテレビ番組が映ってしまうことなんです。

ということは、もしチャンネルをあわせたときに画面全体が真っ黒な放送があるとして、そこでスーパーインポーズしたらどうなるか。……そうですね、コンピュータ画面の黒（透明色）の部分にビデオ信号の黒が入って、うまくコンピュータ画面がそのまま録画できるわけです。

ところが、そんな放送があるわけがないので、どうやって黒色の画像を手に入れるかが問題となってきます。しかも、それを通してコンピュータ画面を見るのですから、ノイズの多いビデオ信号だと録画したときに画像が乱れて見にくいかもしれないので、できるだけ安定したものを探すことになります。

私の知っているものではセガマークIIIやメガドライブ、PCエンジンなどのゲーム機のカセットを入れずに電源を入れると、真っ黒の画面が流れたように記憶しています。ただし、これらは正確にはビデオで使っているビデオ信号とは微妙に異なる場合があるので、もしかしたら同期がずれたりノイズが出る可能性もあります。結局は手持ちのビデオ機器との相性次第ですので注意してください（録画側のビデオデッキにTBC機能がある場合はTBCをON/OFFして相性を調べてください）。

また、2台以上のビデオデッキがある場合、ほとんどのビデオデッキが外部入力にして画像を入力しなければ、画像出力側には真っ黒（灰色？）な映像信号が流れると思います。それらの出力をカラーイメージユニットのビデオ入力につなげておいてスーパーインポーズすれば、うまく録画できるでしょう。

なお、近日発売が予定されているビデオボード（カラーイメージユニットの録画専用版、イメージ取り込み機能はない）では内部にビデオ信号発生機を持っているのでこのような面倒な操作は必要なくなったようです。すでにカラーイメージユニットをお持ちなら特に必要ないと思いますが。



Oh!X1988年9月号のturbo RAY TRACERが動きません。リストを同封しますので、おかしなところがあれば教えてください。

北海道 村松 良彦



村松さんの質問は便箋2枚にわたる長いものだったので、質問を簡略化させていただきました。

ところで、送られてきたリストと質問の内容から判断すると、こちらの説明不足のため動作していない可能性もありますので、一応補足説明させていただきます。

記事ではリスト6からリスト9がデータの例として掲載されていますよね。これらのデータはリスト3のデータセットプログラムにマージして使うようになっているのですが、そのことが記事の中で触れられていません。たとえば、リスト6の例1が「EXAMPLE1」として保存してあるのなら、リスト3をロードしたあとに続けて、MERGE「EXAMPLE1」

のようにするのです。RUNすると、

INPUT FILE NAME :

と表示されますが、それにはリターンキーを押すだけで結構です。

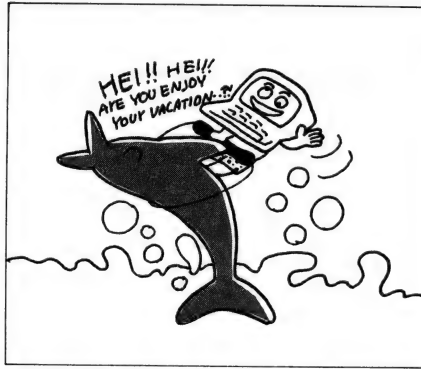
これで駄目ならプログラムに入力ミスがあるものと思われます。（影山 裕昭）

質問にお答えします

日ごろ疑問に思っていること、どんなことでも結構です。どんどんお便りください。難問、奇問、編集室が総力を上げてお答えいたします。ただし、お寄せいただいているものの中には、マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限、マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名、システム構成、必要なら図も入れてこと細かに書いてください。また、返信用切手同封の質問をよく受けますが、原則として、質問には本誌上でお答えすることになっていますのでご了承ください。なお、質問の内容について、直接問い合わせることもありますので、電話番号も明記してくださいね。宛先：〒108 東京都港区高輪2-19-13

NS高輪ビル

ソフトバンク株式会社出版部
「Oh!X質問箱」係



FROM READERS TO THE EDITOR

もうすでに、外は夏真っ盛りかと思うほどの暑さ。でも、編集部の中はクーラーが効いているからすずしい、……はずな

の다가調子が悪いのか、なんか暑い。仕事なんかできな一い。というわけにもいかず、しかたなく働くのであった。

◆本屋で「Oh!X」を手にしたとき、やっとディスク付録がついたかと思った。レジで「780円です」といわれて700円しか持っていなかった僕はスクーターをとばして80円を取りに戻った。ディスクを立ち上げて苦労しただけのことはあったと思った。 能美 和具(19)熊本県

すいませんねえ。しかし、予告で780円になりますと断ってあったとは思んですが。5月号は立ち読みもしなかったのかな。

◆今月号は付録つきのために異常に薄い! と思って購入したら「Oh!PC」ではなく「Oh!X」でした。貴誌の迫力(?)ある誌面に洗脳され我が家配備第1号はPC-286シリーズのハズがX68000PROIIになりました。今後のご発展を祈ります(期待してます!)

坂本 慎太郎(29)東京都
やった、偶然とはいえ読者が増えた。しかし、Oh!PCのほうでひとり減ったのかな。同じ会社の雑誌だからなあ。

◆半年に1回ぐらいはディスクをつけてもパチは当たらないと思う。朝野 貴敦(17)滋賀県
いや、パチが当たって倒れる人が続出して本が出なくなるような気がします。

◆3週間に1度ぐらいディスクつきにしてほしい。 野田 佳照(16)愛知県
??????????

◆今回のディスクは僕の人生に常駐した。

中島 潤史(15)埼玉県
早くワクチンを使って頭の中を治療しないとボロボロの人生になってしまうかも(どういう意味なんだ)。

◆創刊8周年なんですね。ちなみに私のX68000PROは1歳です。だから、誕生日(買った日だよ)には1Mバイト増設してやりました。

山田 雅宏(18)岐阜県
このプレゼントで幸せになれるのはX68000か、あるいは本人か? たぶん両方ともにでしょうね。

◆6月は創刊8周年記念だということを知って自分の誕生日も6月なのでうれしかった。

佐藤 直人(11)神奈川県
実は編集部のE.O.さんも6月に誕生日を迎えたので、みんなでケーキを食べました。ひさびさのケーキはおいしかったなあ。

◆今回のようなディスク配布が不可能であれば、ダンプリストを圧縮した形で載せてもらいたい。入力が楽である。 塩谷 隆治(31)広島県
そうですか。じゃあ、来月からリストを1文字=0.1mmぐらいの大きさに圧縮して載せますので、皆さん虫眼鏡を用意して待っていてください(冗談)。

◆いまYet Another Columnにはまっている。最初の頃は2時間ぐらい座りっぱなしだったが、いまでは自分の力量がわかってしまってたか、2回ぐらいでやめてしまう。編集部には4万点をこすつたものがあるそうだが、どんな手を使っているのでしょうか。正当なやり方で取れるはずはないのだから、と思う。柴崎 誠(17)福島県
正当なやり方ですよ。どんなやり方かというところ……。

◆編集部での40,860点とゆーのは信じられない。おそらく、「必殺! ESCキーで止めればどこに落とせばいいかわかるじゃないか攻撃!」を使ったのだから、と思う。ちゃんければ、古代フェ



吉田 里志 宮城県
えっ、バックマニアって製造中止になったんですか。本当ですか。ところで、あれとイースIIIって全然違うんですが、忘れられるんでしょうか。

ニキアに生きた者の血を引いてるとか。

大島 貴成(17)栃木県

実はですねー、あのゲームには「なるべく高いところから落としたほうが点が高いんやでー攻撃!」というのがあるのですよ。気づいた人もいますよ。

◆Yet Another Columnは面白かった。ブロックがくずれる音が「ケセランパサラン」と聞こえてしまうのは僕だけだろうか。そして、このゲームを「ケセランパサラン」と呼んでいる。友達におかしいといわれたが、なにがなんでも「ケセランパサラン」と呼んでいる。

奥村 真明(17)埼玉県

僕も自分のイスを「ギシギシくん」と呼んでかわいがっています。また、その友達には時計の「コチコチくん」というのがいます。

◆やられてしまった。かねてから作ろうと思っていた「この木なんの木」。日立社員のこの私が作ろうとしたのに会社が忙しくて……。P.S.習志野工場のお昼にはかかりません。さみしいなあ。ほかでは確かにかかるところもあるのに。ところで、一般人には「この木なんの木」が日立の社歌だと思っている人がいるようだが、これは社歌ではありません。社歌は別にあります。念のため。

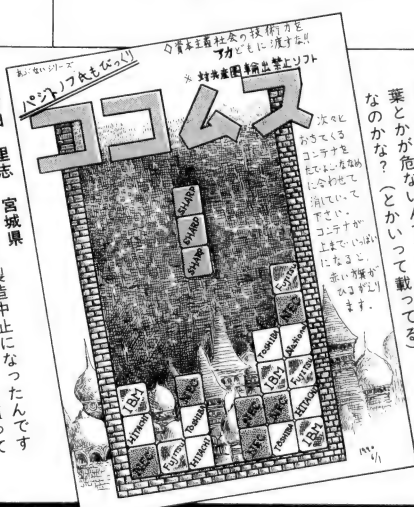
藤井 実(19)千葉県

アンケートハガキを読んでも「この木なんの木」がよかったというのが予想外に(?)多くてびっくりしました。別に「よかった」というのが予想外なのではなく数が本当に多くて。

◆Oh!Xを買いはじめて7,8カ月。そろそろ内容のベースにもついていけるようになりました。最初は内容についていけず(突然OPMAだとか書いてあったので)とても困り、なんて不親切な本なんだろうと思いましたが私もいいかげんなもので、いまではもっといろんなツール載せろーと心の中では思っています。これからいろいろなプログラムを載せてください。次回の付録ディスクが近いうちにあることを願っております。では、さようなら。

小田 典央(19)静岡県

まあ、人間ってそういうもんですよ。



大野 真実 静岡県
あぶないシリーズ第2弾です。しかし、本当に言葉とかが危ないんですけど……。載せても大丈夫なのかな? (とかいって載ってる)

◆もっと本を厚くして5月号に載っていたX68000の変なデモみたいのをたくさん載せてもらいたい。

望月 伸幸(17)静岡県
変なデモというのはひょっとして(て)のショートプロバートに出ていた例のアレのことでしょうか。本が厚くなってあいうのばっかり載っていたらとてつもなく恐ろしいような気がします。

◆うーん、なにか押し入れの中でカサカサ音がするなあ……、と思って押し入れ開けてゴソゴソやっていたら、「ゲッ!」、思っていたとおりゴ、キ、ブ、リとご対面してしまった。予想していたこととはいえ、やっぱり気持ち悪い。と、躊躇していたらフトンの中に逃げ込まれてしまった。おそらく、まだ中にいると思われます。田舎にいた頃は東京近辺よりは湿度が少なかったせいか、ほとんどゴキブリは見たことがなかったのですが……。ゴキブリを見かけたせいで、「ああ、俺も関東に住んでいるんだー」と妙に感激してしまった。が、やっぱりイヤだなー。でも、早めにやつけんといかん。うん。

工藤 隆(20)埼玉県
ゴキブリのもっとも恐ろしいところとは……。それはやはり、叩き殺そうとしたら顔に向かってバタバタと飛んでくるところでしょう。あの瞬間のこわさときたらこの世で1番じゃあないかと思ったりします。

◆僕の友達が考えた“パソコンとカツ丼を手に入れる方法”。

- 1) 展示パソコンを持って逃げる
- 2) 逃げる途中に隠れている友達にパソコンを渡す
- 3) わざと警察に捕まる
- 4) 黙秘権を使う
- 5) しばらくすると警察がカツ丼をくれる
- 6) もうちょっとすると釈放される

これでパソコンが手に入りカツ丼も食える。すごい!

小川 伸一郎(15)京都府
いやー、すごいですね。15歳(?)にしてこの頭脳。編集部一同思わず感心してしまいました。まさに完全犯罪ですね。ひょっとしてノーベル賞ももらえるかも。どうもおめでとうございます。

◆X68000も10万台をこえたようなので、そろそろマニア以外にも売れることを考えたらどうだろう。案としては自己診断機能の高度化。たとえば、まずコンセントを入れると周辺機器をチェックする。ディスプレイやキーボード、マウスが接続されていないと、「私の顔をつけて」とか「私のねずみはどこ」と話して誰でも接続できるようにする。

笠井 康彦(23)神奈川県
すると、接続を間違えたりすると「そこじゃないわよ」とか、スイッチを切ろうとすると「やめて」とかしゃべるのだろうか。あー、気持ち悪い。

◆PC-9801と同じくらい普及しているビジネスパソコンであるといつてX68000を買ってもらったのに例のウイルス事件によってうそがば



杉本 秀昭 宮城県
なんかよくわからないけど、とまかくそんな格好していると風邪をひきますよー。とかなんとか、いい



大村 直人 北海道
どうもありがとうございます。やはり、こういうお祝いのハガキもないとね。しかし、とっても味わいのある(?)イラストですね。

森下 剛(14)京都府
そんなすぐにばれるようなうそを……。

◆最近、アクションゲームやロールプレイングゲームに興味がわなくなった。どうしてだろう。

中井 卓(18)大阪府
どうしてだろう。きっと大人になったんだ

よ。
◆涙の浪人生活に入ってから小遣いを1,000円に減らされてしまった。しょうがないので弁当を作ってもらえなかった日に食事をぬいて300円ほどひねりだし、やっとOh!Xと好きなバイクの雑誌を買っている有り様。なんとも情けないことであります。しかたないですけどね。最近ではゲームもあんまりしていなかった(というより、「これ!」と思うのがなかった)ポピュラスを知ったとき、はまってしまいそうでこわい思いながら金がないのでさみしく思っていました。そこにこのプレゼント。僕にポピュラスをくれー。Oh!Xを買始めて7年目。小学生だった僕もいまは浪人生、なんかすごいものを感じるなあー。

安陪 亘(18)三重県
ううっ、なんて情けない。ごはんを抜いてその浮いたお金でなにかを買うというのはよくある話ですが、体をこわさない程度にしましょう。でも、そうかといってポピュラスをあげるわけにはいかない。

◆ふと思った。ファジィコンピュータ内蔵(ぢゃなくて内蔵)のカメラで撮った写真はどのようになるのか。

fuzzy(形)「中略」 2. [写真が]ぼやけた(blur red) シニア英和辞典 4訂版より

大村 直人(17)北海道
なるほど。

◆暑さが厳しくなってきたなか、部屋に閉じこもりっぱなしだと頭がどうにかなりそうです。懸賞にクーラーもつけてください。

荻久保 雅道(14)静岡県
僕もクーラー欲しい。

◆以前、続けて4回足を運んだ映画のサントラ盤をステレオを持っていないのに買ってしまっ、そのレコードのためにステレオを買ったことがありました。今月号の付録のディスクを見てふと思い出してしまいました。

三原 克之(36)福岡県
そういえば、僕もCDラジカセしか持っていないのにレーザーディスクのソフトやレコードを持っている。

◆X68000が10万台前後だそうですが、もし個人でソフトハウスを開業したとして1パーセント以上の人が(通信販売で)ソフトを購入すれば経営が成り立つと思います。「私はやってみたい!」と思っている人はかなりいるのではないのでしょうか。ですから、ソフトハウス経営についての特集をお願いします。特に、ダビング工場のメーカー名と連絡先やその手数料、パッケージの単価と依頼数量など。この特集をすることにより、X68000ユーザーの中からソフトハウスを開業する人が多く出る→ソフトが増える→X68000購入者が増える→Oh!X購入者が増える!

高久 裕明(29)東京都

やはり、問題はその個人が作ったソフトが市販ソフトとして受け入れられるようなレベルに達しているかどうかでしょう。つまらなければ、やっぱり全然売れないだろうし、面白ければ販売しようという話はどこから来るでしょうから。

◆ANGELの人体モデルはどうして女の人ののですか。

竹永 昌伸(16)兵庫県

うう、それだけは聞かないで。じゃなくて、ただ単に男だと気持ち悪いからじゃないでしょうか。

◆アンケートハガキの何パーセントが読まれているのだろうか。読まれないのであればなにを書いても出さないと同じじゃあないか……。

小杉 雅信(21)愛知県

全部読んでるに決まっているじゃないですか。このコーナーやハミダジっていうのはアンケートハガキによって成り立っているんですから。だから、白紙とかでなくなんか面白いことを書いて出してください。スタッフの人なんかにもくるたびにハガキを読んでもすよ。

◆いつもOh!Xの記事を見て、すごくらやましくなります。なぜかといえば、SHIFT BREAKとかmicroOdysseyとかみたいに自分の考えを自由に(多少は制限があるでしょうが)書いて、ま

たそれに対して読者から意見がきて、またそれに対して意見を言えるという。なんか、そういうのっていいですね。いちばんうらやましいのはやっぱり「STUDIO X」の答える人かな。一度でいいから代わってほしいと思うのは僕ぐらいなものでしょうか。 斉藤 哲哉(18)愛知県
そんなにうらやましいですか? まあ、一応仕事としてやっているんですが、確かに自由に書いたり、その反応が返ってくるというのは実に楽しいことです。

◆気がついたら、知らない人の家にいた。大学の芝生の上に寝ていた。梅田の映画館の中にいた。先輩、日本酒とビールのカクテルの中に味の素、塩、魚の頭、キャベツ、しょう油を入れて飲ませないでほしいな(文科系サークルとは思えないところに入った……)。

佐藤 能久(19)大阪府
いや、体育系より文科系のほうが飲み会がきついというのはよくある話ですね。しかし、魚の頭やキャベツだったらいいますよ。もっと、ひどい話を聞いたことがあります。それは、……(あまりにもひどくていえない)。

◆X68000のスーパーインポーズでそのニュース速報(チャイムつき)を流し、バアさんを指名手配の犯人に仕立て上げたら、バアさん3日間悩んだ。 松本 浩一(24)栃木県

僕もそういうことを考えてPC-6601SRでやろうと思ったのですが、グラフィックが粗いので漢字がでかくなるし、第一、専用ディスプレイがなくてスーパーインポーズができなかったのです。ううっ、悲しい思い出なあ。

◆ゆるせないぜ! アンケートハガキの下の“X68000(無印、ACE、PRO……)”の無印のはなんだよー。初期型はな、グラディウスが付いてたんだぞ。CZ-600C万歳!

御厨 桂治(18)山梨県
何をいってるんです。無印良品っていうじゃないですか。うーん、しょうもない答えになってしまった。

◆いま気がついたのですが、アンケートハガキの裏表に年齢を書く場所があるのには意味があ

るのだろうか(すでにどなたかが気づいているかもしれないが)。もしかして、裏の年齢は愛機の年齢を書くのだろうか。

西谷 健吾(17)兵庫県
違います。裏には数え年を書くんです(またまた、しょうもない答え)。

◆HDタイプのX68000は地震に弱いので対策を立てました。それはキャリングハンドルを利用して天井からロープで吊るすのです。そうすれば、ソバ屋の出前バイクの法則によりX68000は地球の重心に対して静止するのでクラッシュの魔の手から逃れることができます。ぜひ、おためしください。それにしても大洋は強い。

矢地 雄(18)東京都
部屋が広ければ問題はないけど、せまかったらロープの長さによっては悲惨なことになるそう。壁にぶつかって。そうでなくても、落ちたときのことを考えると、とてもおためしなんかできない。

◆なんということか。「ハード」のプレゼントがないじゃないか! 私は楽しみにしていたのに(当たるわけでもないけど……)。今月号はX68000を持ってればとってもうれしいのかもしれないが、ほかのユーザーはどうしろっているんだ。

秋友 謙二(16)山口県
「ハード」のプレゼントは今月だったんですよ。はっはっは。しかし、なかなか当たるのは難しいでしょうね。

◆読者の方に聞きたいんですけどマウス、トラックボール、みんなはどっちを使っているのでしょうか。私の場合、部屋が狭い(4畳半、バス、トイレ、キッチン共同で家賃8,000円。今春から1,500円上がった。くるしー)ので机の上にキーボードとサイバースティックを置くといっぱいになり、マウスとして使うスペースがなくトラックボールとして使っています。両手はふさがりますが、そのぶんマウスのときのような腕の筋肉痛(あるわけねー)がなくなります(運動量が少ない)。みなさんはどっちです。

栗 幸司(21)広島県
僕はマウスとして使っていますが、机の上の空きスペースが10×10cmぐらいしかないので非常に苦しい。

◆初のフロッピーディスクの付録、年寄りには最高のオマケでした。長いリストを打ち込むことは体力が持ちません。最近ではリストを見るだけであきらめていたものでした。年寄りのためにもこれからときどき入れてほしいと思います。

小池 清(42)滋賀県
年寄りというほどの年でもないと思うんですが。まあ、長いリストを打ち込むのってけっこう体力が必要ですね。

◆愛読者年間モニタの応募者が欠員というのは、とても残念です。読者の皆さんがどうせなれないだろうと敬遠しているのか、本当に参加意識が薄れているのかはわかりませんが、7名というのには驚きです。私は第1期のモニタをさせていただいたので前者のほうですが、モニタ経験のある者でももう一度できるものならぜひやりたいところです。たぶんあの記事に刺激されていまではかなりの数の応募があると思います

が……。 紺谷 憲児(22)大阪府
別に一度やったからといって、年間モニタが二度とできないということはありませんから、経験者の方もどんどん応募してきてください。

◆1年ぶりにXIturboと再会した。が、2,3回スペースキーを叩くとスペースキーが死んだ。こうなるとほとんどのゲームができない。しょうがないのでワープロとして無理に使っていた。でもこれでは面白くないので、近くの電器屋に修理に出したらキーのスイッチとコールコードの交換で1万円以上もした。おかげで翌日のビジネスショウに行けなくなった。しかもである。スーパー大戦略をやっている気がついたのだが、HELPキーが死んでいる。どーしよう。あんまり使うキーでないだけに悩んでしまう。

加藤 健二(18)埼玉県
まさに「一難去って、また一難」。

◆やっぱりX68000はいいですね。あつ、そういえば4月のいくくんだったか忘れましたが、夜、MOTOSを立ち上げたらいつものオープニングの曲と違う曲が流れたんです。あれは、なんだったんでしょう。 野口 智広(17)神奈川県
さあ、なんだったんでしょう。

◆バットモービル届きました。こんな凄いプレゼント生まれて初めてです(笑)。とりあえずディスプレイの上に飾ってあります。暇になると走らせてみたりしていますが、傍から見るとちょっとあぶないやつに見えるかも(かもじゃないって)。

松久 孝治(20)岐阜県
走らせるときに「ブーン、ブーン」とかいうと、なかなかいいかもしれない(なにがいいのやら……)。

◆はじめまして。僕はX68000を買って(もらって)1年と少したちました。買って1カ月ほどたってから今まで、BASICを興味だけで学んできました。自分ではなかなか進歩したと思って、そろそろ高レベルの雑誌を購入しようと思いOh!Xを買うにいったのです。が! 内容を見たとなん、全身の血が凍ったかと思うほどにおどろいた(なんじゃそりゃ)。今まで僕がコソコ

Super Intelligent Book series.

Oh!X 別冊

清涼飲料水 全カタログ

あなにはここに掲載されている全ドリンクを制覇出来るか。

古村 聡 著

SOFT BANK

清水 健年 東京都
こんな本が出たら売れるんだらうか。うーん、やっぱり結構売れてしまうんだらうな。しかし、読んだら飲んでみたくなるんだらうな。

MAKE YOUR LANDS POPULOUS

信川 洋 東京都
ポピュラスって本当に人気が高いんですね。シナリオオデッセイのフロミストランドも出たことだし、ますますハマル人が続出するのかな。

ツ学んできたことは、まるで宇宙の中の人間……まではいかないが、星のような（あんまし変わらん）ものだったのです。そんで、今月号の中に「PROを買って半年たって……」などといったが僕にはわけのわからんことが書いてあったりします。いったいこーゆー人はどうやって学んだのか……。教えて！

田村 高志(16)愛知県
人は人、自分は自分ですから、マイペースでコツコツやるのがいいんじゃないですか。試行錯誤しながら自分で学んだほうが身につくし、面白いですからね。

あとがき

ここに載っているのは6月号のハガキからなんですが、6月号のアンケートハガキはやはり、ディスクに関するものが多かったですね。目にしたものをちょっとまとめてみました。

・Yet Another Columnが面白くてはまった（定期試験があるにも関わらずとか、忙しいのにとかいうのが多い）。

ちなみにハガキで書いてきた中でのハイスコアは滋賀県にお住まいの小野さんの50,875点で

した。

・ディスクつきになるんなら、〇〇円出しても買う。

・ディスクをフォーマットしてしまった。

こういう人が何人かいましたが、やはりプロテクトシールを貼っておいたほうがよかったのかな……。まあ、そういう場合はしょうがないのでそのディスクをこちらへ送ってもらえればもう1回書き込んで送り返してあげられると思います。

・6月号は保存版にしたいので2冊買った。

・ANGELが走らない。

ANGELはまったく走らないと思っている人が多いようですが、そんなことはありません。

勘違いをしている人がいるかもしれませんが、6月号の49ページの「2）FLOAT2+」の動作がおかしくなることがある（たまたま計算を間違う）。ANGELは実行しないように。」というのはFLOAT2+、Xを使ってANGELを実行しないということではありません。FLOAT2+、Xを使えばANGELはちゃんと動きます（ANGELに



▲熊谷 逸郎 京都府

なんでフラッピーがスブラッターハウスに……。なんで似合ってますよね。マスクを取ったときに爆笑されそうだけど。

は多少おかしいところもありますが、起動は絶対にできるはずですよ。走らない人はもう一度本文をよく読んでください。

なんか違うコーナーのようになってしまいましたが、今月は大部分がディスクに関するハガキでしたので、あしからず。

ぼくらの掲示板

仲間

- ★「バ〇レイバー」のサークル「狂走都市」ではCGを大々的に扱っている会報を発行中です。CGのPDS発行も企画しています。「バ〇レイバー」だけでなくほかのゆうき先生の作品をCGで描いてみたい15歳以上の方、入会してください（当会所有機種はX68000 EXPERT、PC-9801RX21）。62円切手同封でご連絡ください。〒949-66 新潟県南魚沼郡六日町大字六日町484 種村聡人(15)
- ★X68000ユーザーを対象とした(X1でも可)サークル「X68K-GR」の会員を募集しています。ゲームを中心に月1回程度の会合を開き、ゲームの情報交換や会誌の発行をしたいと思います。興味のある方は簡単な自己PRと62円切手同封のうえ、下記の住所までご連絡を。〒239 神奈川県横浜須賀市ハイランド5-3-19 三浦正義(17)
- ★「ディスクサービスX68000」。このサークルではX68000の機能をより高度に活用するためにユーザーの方を募集しております。活動内容は皆さんから集めたプログラムを会報にまとめ、配布するというものです。興味のある方は62円切手を同封のうえ、ご連絡ください。〒165 東京都中野区上鷺宮5-28-24 前野千絵(21)
- ★このたびサークル「べけるく亭」では会員増強に伴い第3期会員を限定募集いたします。パソコンをより高度に活用するための情報交換などが主体です。入会希望の方は62円切手同封のうえ、下記までご連絡ください。〒491-03 愛知

- 県一宮市萩原町富田方茶原54 野村真広(23)
- ★このたび、X68000ユーザーのサークルを作るにあたりましてプログラマを募集しようと思います。できればアセンブラやC言語のできる方、興味があってこんなボクでも力になれるなら……とか思った人はぜひご連絡を。〒437-11 静岡県磐田郡浅羽町浅羽1169-32 袴田信孝(17)

売ります

- ★MZ-2000用周辺機器を以下の価格で（送料込み、値引き可）。フロッピーディスクドライブ「MZ-1F07」を2万5千円、プリンタ「MZ-80P6」を1万円。いずれもインタフェイス、ケーブル、説明書つき。拡張ユニット（MZ-1U01）、漢字ROMボード（MZ-1R13同等品）、VRAM 3ページを各8千円で。連絡はハガキで。〒371 群馬県前橋市上小出町6-1-205 中嶋康弘(31)
- ★X1用データレコーダ「CZ-8RL1」を5千円（送料込み）で。連絡は往復ハガキで。〒737 広島県呉市阿賀中央1-24-20 清水幹雄(18)
- ★MIDI音源ローランド「D-110」（1年使用）をマニュアル、保証書、付属品一式、1Uラックつきで4万5千円で譲ります。送料当方負担。連絡は往復ハガキで。気長に待ちます。〒594 大阪府和泉市鶴山台4-8-3 山路智弘(20)
- ★IO-735X（カラーイメージジェット）、マニュアル、付属品、X68000用ケーブルつき、箱あり、キズなしを送料込み8万円で。連絡は往復ハガキで。〒238 神奈川県横浜須賀市深田台76 小林秀樹(21)

- ★ワープロ書院「WD-540」を5万円で。箱、マニュアル、付属品、保証書あり。おまけでインクリボン、フロッピーディスクをつけます。連絡は往復ハガキで。〒187 東京都小平市花小金井4-286 小川和幸(24)
- ★増設RAMボード「CZ-6BE1」を送料込みで2万2千円程度で。少しぐらいなら割引します。マニュアル、付属品あり、キズなし。連絡はハガキか手紙でお願いします。〒665 兵庫県宝塚市南ひばりが丘3-26-5 関口敬文(15)

買います

- ★X1用FM音源ボード「CZ-8BS1」（完動、箱、マニュアル、付属品あり）を送料込みで8千円から1万円で。連絡は往復ハガキで。〒673 兵庫県明石市西明石北町3-16-5 春名隆行(17)
- ★MZ-2521用辞書ROM、増設RAMボード、増設VRAMボードを各6千円ぐらいで。コンパチ品も可です。希望価格を明記してハガキで。〒458 愛知県名古屋市中区鳴海町神明163-1 安川実(16)
- ★X68000用増設RAM「CZ-BE1」を1万5千円ぐらいでお願いします。完動で付属品つき。連絡はハガキで。〒285 千葉県佐倉市井野869-26 松本琢磨(17)

バックナンバー

- ★Oh!X1989年5月号を千円、「X68000テクニカルデータブック」を2千円で。美品希望。連絡はハガキで。〒289-13 千葉県山武郡成東町成東2470 安井忍(22)

▶最近このコーナーで紹介しているサークルに対して、問い合わせでも返事がこない、他人の著作権を侵害する行為が行われているなどの残念な報告が増えてきました。信頼関係に基づくコーナーですからマナーの向上に留意してもらいたいものです。今後、仲間のコーナーでの紹介を希望されるサークルは必ず会誌の見本を送ってください。

DRIVE ON

このコーナーでは、本誌年間モニタの方々のご意見を紹介しています。今回は6月号の記事に関するレポートです。これまでのモニタの方にとっては最後のレポートです。1年間のモニタレポートご苦労さまでした。

●“共通システム”という考えに基づいたS-OS思想は、5年たったいまでも決して古びてしまったものではない(それどころか、いまだからこそ重要なことかもしれない)。しかし、“SWORD”というシステム自体は8ビット全盛時代のものでしかない。16ビット以上の時代の8ビットのためのシステム、“Excalibur”なり“Storm Bringer”なりを発表すべきではないだろうか。INTEGRAL XI (KAME-DOS)のように、どんなディスクフォーマットも読める機能やヒストリといった近代的機能を備えるS-OSが発表されてもよいと思う。だからといって、テープユーザーや旧機種ユーザーを切ってもいいというわけではないが。

西田宗千佳(18) X68000, XIFmodel 20 千葉県
●なんといっても“PurePASCAL”, これに尽きるのではないのでしょうか。グラフィック機能が標準でないのが残念ですが、これが付けばかなりなものになると思います。私のようなPASCAL派はX68000ユーザーの中には少ないかもしれませんが、これを機にPASCALを勉強してPASCALの素晴らしさを知ってほしいと思います。できれば、コンパイラの内部仕様などに関する記事があってもよかったのではないかと思います(まあ、これは今後に期待しましょう)。

森川一(24) X68000ACE-HD, Xlturbo II 北海道

●「ハードウェア工作入門」についてですが、製作の対象とする回路はできるだけシンプルなものをお願いしたいと思います。また、なるべくローコストでということも。最近では気軽にハンダゴテを握ったり、紙工作したり、プラモデルを作ったりというような話をあまり聞かなくなりました。これはやはり、身の周りに完成品があふれているためでしょう。でも、誰でもみんななかしらの創作意欲を持っているはず。ハードウェア工作入門には、そんな私たちの創作意欲を刺激し満足させてくれる連載になってほしいと願います。

藤田康一(19) X68000PRO 静岡県

●S-OSがまさかX68000やPC-286にまで広がるとは思っていませんでした。うれしかぎりです。でも、X68000ユーザーはともかくとして、PCユーザーがこのことを知らないのは残念だと思います。Z80シミュレータとしてなんとかPCユーザーに知らせる方法はないでしょうか。ちなみにPC-9801RSで動かしてみましたけどXIよりも少々速いような気がしました。なんといっても2Dのディスクの読み書きができるのは5重丸です。

末吉克行(21) XIG, MZ-731, FM-7 兵庫県

●「ハードウェア工作入門」ですが、前回のアンケートで書いたことはちゃんと押さえてあり、「何が必要であるか」ということがわかりやすく書いてありました(さすがだなあ)。プログラムのように、簡単にはやり直しが効かないハードウェアが相手ですから、なかなか大変だと思います。入門講座の場合いちばん大切なのは、「急に難しくならない」とことだと思います。余談になりますが、NHK基礎英語がいまだに入門講座として利用される

ことが多いというのは「急に難しくなることがない」からなのだそうです。そうしないと、ついてこれないというわけですね。バカ丁寧すぎるくらいいいですから、ゆっくりのんびりやってほしいですね。それと、なるだけわかりやすい図を使ってほしいと思います。ジョイスティックポートにつなぐものがほとんどと聞いて安心しました。実際に組み立てる場合、回路図と配置図がバツとは結び付かないものです。毎回言っていることなのですが、「難しいことはば、脚注などを付けてもらいたい」と思います。X68000マシン語講座がなぜ読みやすいかというと、脚注などが詳しく、難易度に気を配っているからだといえます。ハードとソフトの違いはあるとはいえ、やっぱりこうあってほしいと思います。湯澤聡(27) X68000, XlturboIII, MZ-2861/2531, PC-6601, MSX, PC-1360K 埼玉県

●アンケート結果を見て。やってくれますねー。まあ、そうとうの内輪ネタであるのですけれど、こういった内容であればいたしかたないでしょう。さすがのX68000の伸びと、ほかの項目内のX68000が占める割合が、いまいばん私にとってショックですね。あのとき、XlturboZがX68000か多少なりとも悩んだんですから。ベストライター(もちろんOh!Xのスタッフ1人ひとりとはベストライターです)、なんていうアンケートはまさか載せるためだとは思いませんでした。祝一平氏がトップなのは、やはりという感じ。ま、ほかにもいろいろありましたが世論調査みたいでいいですね。作り手と受け手がこうもコミュニケーションできるのはOh!Xだけでしょ。また、やりましょう。

大津和之(20) XlturboZ 福岡県

ごめんなさいのコーナー

7月号 AFTER REVIEW

「サーク」のレビュー内の写真が「ルーンワース 黒衣の貴公子」のものと入れ替わっていました。関係各位にはご迷惑をかけました。お詫びいたします。

7月号 WZD

先月号のものではコマンドラインからパラメータ付きで実行した場合、復帰時の動作が保証できません。詳しくは今月号のP.147をご覧ください。アセンブル時は問題なく動作するはずですが。

6月号 ANGEL

P.65 回転のコマンドの書式に間違いがありました。

rotx <式> → rot.x (<式>)

roty <式> → rot.y (<式>)

rotz <式> → rot.z (<式>)

のように変更してください。お詫びして訂正いたします。

また、画面をはみだすような絵を描かせることになってしまうことがあるようです。

6月号 GCC Ver.1.36.01

Humanのバージョンが2.00の人はうまく動かないようです。前にも書いたようにHuman v.2.00はシャープでv.2.01に交換してくれまので、これを機にバージョンアップしましょう。

6月号 X68000マシン語プログラミング

files.hのリストが抜けていました。詳しくは今月のX68000マシン語プログラミングをご覧ください。申し訳ありませんでした。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日 16:00～18:00

お問い合わせは原則として、本誌のバグ情報の方に限らせていただきます。入力法、操作法などはマニュアルをよくお読みください。また、よくアドベンチャーゲームの解答を求めるお電話をいただきますが、本誌ではいっさいお答えできません。ご了承ください。

本誌創刊号 をプレゼント！ そんなバカな〜っ

▼お陰さまで本誌は通巻100号を迎えることができました。今月号は、グラフィック特集、表紙ぎゃらりいなどでカラーページを増ページして豪華にお送りいたしましたがいかがでしたでしょうか。

さて、編集部ではこれを機に取り置ききのバックナンバーを整理し、その一部をなんらかのかたちで皆さんに提供したいと考えています。とりあえず今回は100号記念プレゼント番外編として、Oh!MZの創刊号を3名の方に差し上げたいと思います。ご希望の方は縦じ込みのアンケートハガキのプレゼントNo.1に0と記入してお送りください。

▼本誌では、コンピュータサークルなどの制作による同人ソフトの紹介を考えています。特にX68000などのユーザーグループの作品にはレベルの高いものが多く、市販ソフトにはない手作りの味が魅力です。これらはパソ

ケットなどを通じて安価に販売されていますが、一般にはあまり流通していません。本誌ではこうした作品を広く読者の皆さんに知ってもらいたいと思います。本誌での紹介を希望するソフトがありましたら、編集部までご連絡ください（☎03-5488-1309）。また、団体名、連絡先、代表者名を明記のうえサンプルソフトをお送りいただければ幸いです。

▼ここで嬉しいお知らせです。しばらく本誌を離れていた清水和人氏が次号より復帰。ゲームやプログラミングの楽しい記事をお願いすることになりました。ご期待ください。

▼先月号でお知らせしたとおり、7月1日から株式会社日本ソフトバンクは「ソフトバンク株式会社」と社名を変更しております。また、社屋も移転となり、Oh!X編集部は16日より新しい編集部にて業務を開始しております。お問い合わせの際には、電話番号が変わっておりますのでご注意ください。

▼先月号に掲載した日コン連企画の広告中、XIユーザーに対して不適切な表現があり、ご迷惑をおかけしました。広告主になり代わり、深くお詫び申し上げます。

投稿応募要領

- 原稿には、住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は、詳しい内容の説明、利用法、できればフローチャート、変数表、メモリマップ（マシン語の場合）に、参考文献を明記し、プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また、掲載にあたっては、編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は、詳しい内容の説明のほか回路図、部品表、できれば実体配線図も添えてください。編集室で検討のうえ、製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして、他誌との二重投稿、他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㊟㊷㊸」係

S H I F T ・ B R E A K

▶マーク・トウェインの「不思議な少年」を読んだ。天使「サタン」が人間の矮小な卑俗さを描いてみせる。そのなかに彼が泥の小人を箱庭に生活させ、城を作ったところで雷で皆殺しにするくだりがあった。彼は泥人間の運命なんか気にもしていない。なんて残酷な奴だ。……さて、読書はここまでにしてポピュラスでもやろーかな。今日は272面だ。（H.U.）

▶5月号でレビューした天下統一だが、実はとても速かったのだ！ 愚かにも「もうCPUの速さがでちゃうんだよう」などと書いたが、なんのなんの、製品版は80286のRXにも決してひけをとらない。いや、それ以上だといえる！ こんなところでフォローしても何人の人が読むかわからないが、とにかく！ おもしろい！（亀）

▶出張でスペインとポルトガルに行くことになったのだが、出発予定日の3日前になっても飛行機が何時に成田を出るのか知らされていない。それどころか、旅行会社は昼出発と言い、航空会社は夜出発だと言い張る。おまけに、旅行会社から聞いたポルトガルのホテルの住所は実在しない地名だと言う噂だ。果たして無事たどり着くのだろうか。（K.M.）

▶ポピュラス全500面クリア達成しました！ でも、エンディングのようなものではなく、0面（GENESISのようなもの）に戻ってしまうんですね。もしかして、2周しないとエンディングが見れないとか!?（R-TYPEみたいだな）今度は「対戦」と「プロミストランド」を究めてみようかと思えます。Uさんこの間の勝負は練習ですよ、フフフ。（善）

▶ちょっとお尋ねしますが、皆さんの中でトマトジュースのお好きな方はいらっしゃいます？ 僕はあれが好物なのです（野菜ジュースはもっと好き）が、友人から毒物飲料のごとくいわれてしまいました。健康飲料と称した妙なのがいっぱい出てくるずっと前からあった、由緒正しい飲み物のはずなのに（関係ないか）。誰かが賛同を！（A.T.）

▶以前自分が担当してたころの質問箱のページを読み返してみたら、上手く書いてんだよ、これが。内容、文章ともに完璧に近い。俺って凄かったんだなーって本気で思ったね。つまりいたり、水たまりに落ちたり、犬の〇〇踏んづけたりするのを気にせずに空を見上げて歩いているほうが気持ちいい。ということらしい（おや、雨だ）。（Mu）

▶なれど、高校生の頃から大嫌いだったのが“自然保護”という言葉であった。だって、保護というのは「自分より弱いものを守ること」ではないか。いつから人間は自然を保護できるほど偉くなったんだ？ いつからそんなに傲慢になったんだ？ もっと謙虚になりなさい。謙虚に。そして、正直に「人間保護」とでもいってなさい。私は悲しい。（K）

▶ネット上のジョークを真に受けるヤツ。7月を待たずにウイルス終結宣言をするお役所。しかし、ウイルス学会というのはウイルスを作ろうという学会だったとは……。さて、POPULASの決め手は序盤。2つ目の城を何秒で作るか、いかに海を制し、どれだけ速く侵略できるかにかかっている、と思う。マップのせいにしちゃいけませんよ。（S.N.）

▶新社屋となるNS（日本ソフトバンクではなく、日本食堂の略）高輪ビルへ見学に。下には富士銀行、上にはレストランという結構な趣。ところでその日は変な考えばかり浮かぶ日で、社長室を見ては「6万円ぐらいで貸してくれないかな」とか、帰りにNECのスーパータワー（風穴のあいたビル）を見て「あつ、クレイジークレイマー」とか……。 （A）

▶自慢じゃないがシリーズその2。私はいわゆる霊現象によくあう。台所に鑑武者が出たり、遊体離脱や寝入り端の子守歌なんてのはザラ。予知夢も多いし、デジャヴってやつも日に3回くらいある。ほら、こうやって原稿を書いているのだったであつた気が……。そういや昨日にも、その前にも……。ん？ そりゃ単なる習慣だって。（E.O.）

▶創刊100号。私が編集に加わってから52冊、半分以上になるのか。半年前まで最若手だったのに……。編集部が大使館立ち並ぶ千代田区からお寺の並ぶ泉岳寺へ移転することになった。思えばここも3年半。さらば、武道館、靖国神社、北の丸公園……テキ屋にダブ屋の群れ、50mごとに並んだ警官の列……。さらば白百合学園のセーラー服。（U）

▶おかげさまで6月号は売り切れ店続出、なかには500冊以上売っていただいた書店もある。1年間はバックナンバーが買えるよう在庫を増やしたのだから……。さて、その6月号にゲーム基板の話があったが、文脈上Oh!FMのY氏が基板評価に関わっていると誤解を招く部分があり、Y氏には申し訳ないことををした。この場を借りてお詫びしたい。（T）

microOdyssey

東京オリンピックで日本がアルゼンチンに勝ったときは、まだ私にはサッカーのなんたるかわからなかった。目覚めは2年後のワールドカップ・イングランド大会の決勝で、地元イングランドと西ドイツが同点で延長戦に入り、結局イングランドが4-2で勝ったときだ。勝ち越しの1点は、バーに当たって落下し、ボールは外に跳ねかえったが判定はゴールであった。

抗議する西ドイツ選手たちを静めたのはキャプテンのウヴェ・ゼーラー。次のメキシコ大会の準決勝で再びイングランドにリードされたが、なんとロスタイムにゼーラーはゴールに背を向けたままヘディングシュートを決めたのだ。準決勝はさらに激しいイタリア戦。肩を脱臼したベッケンバウアーがギブスで腕を固定してプレーを続ける姿は子供心に焼きついている。西ドイツはやはり終了間際に同点、延長で逆転。が、再度逆転され、さらに追いつくという歴史に残る死闘の末に敗れた。以来、私はずっと西ドイツの熱狂的(?)ファンを自称している。

ああ、それにひきかえ、なんでこんなに弱いんだろうと悲しくなるのが日本のサッカーだ。日本よりも弱い国なんて世界中さがしてもそんなにはない。それも競技人口からいえば結構大国に属するわけで、「いや日本じゃあまり盛んじゃないから……」と言い訳もきかないのだ。

日本が弱い理由は、1) 技術がない。2) 体力がない。3) センスがない。の3点が基本だが、もっと深い部分、思想的な面で問題があるような気がする。サッカーだけでなく、チームプレーを必要とする球技は基本的にダメなのだ。

たとえば、子供たちのサッカーで、キープ力のある子がドリブルで突破しようとする、その子だけが「1人でやっちゃダメでしょう」と注意を受ける。周りの子供が注意されることは意外と少ないものだ。チームプレーに「力を合わせて、助けあい」というイメージが植えつけられるのはこのときからではないか。

プロのサッカー選手が誰かにパスを出すのは、それが自分にとって「もっともいいプレー」となる場合だ。逆にボールを持たない選手はパスをもらえる状況を作るのが仕事の基本だ。

ボールを持つ選手Aはできれば自力で突破したいし、その自信もある。だが別の選手Bが、いやオレにつなぐのがお前にとってのベストチョイスだといわんばかりに動く。Aは、しかたがない、いったん任せるがリターンをよこしたほうが身のためだ、と前に進む。この駆け引きの結果が真のチームプレーとなる。パスは助け合いではなく仕事なのだ。

また、精神面でも多くの教育的指導は勝負に向いていない。たとえば、日本人が大切にしている根性とか精神力とかいう言葉。なぜか「倒れるまで頑張る」ことを美德と誤解している人が多い。玉砕しても負けは負けなのに、である。一方、西洋の精神力は「最後まで倒れない」ことをさす。なぜなら彼らは勝つために頑張るのだ。この違いは大きい。

ひいきの西ドイツは、今回のワールドカップで82年、86年に続いて決勝に進んだ。だが、これぞゲルマン魂という、逆境を勝ち抜く試合が今大会ではまだ見られない。それだけに決勝戦は波乱に満ちた展開を期待しよう。いまは決勝戦を2日前に控えた7月7日である。(T)

1990年9月号8月18日(土)発売

特集1 日本語を処理するために

特集2 2Dグラフィック続論

X68000にハンディスキヤナをつなぐ

新製品紹介 ビデオボード/C compiler Ver.2.0(?)

新連載 清水和人流プログラミング道場/荻窪圭「大人のためのX68000」

全機種共通システム

ビリヤードゲーム

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312	神奈川	厚木	有隣堂厚木店 0462(23)4111
	//	書泉ブックマートB1 03(294)0011		平塚	文教堂四の宮店 0463(54)2880
	//	書泉グランデ5F 03(295)0011	千葉	柏	新星堂カルチェ5 0471(64)8551
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660		船橋	リプロ船橋店 0474(25)0111
	八重洲	八重洲ブックセンター3F 03(281)1811		//	芳林堂書店津田沼店 0474(78)3737
	新宿	紀伊国屋書店本店 03(354)0131	千葉		多田屋千葉セントラルプラザ店 0472(24)1333
	高田馬場	未来堂書店 03(200)9185	埼玉	川越	黒田書店 0492(25)3138
	渋谷	大盛堂書店 03(463)0511		川口	岩瀬書店 0482(52)2190
	池袋	リプロ池袋店 03(981)0111	茨城	水戸	川又書店駅前店 0292(31)0102
	//	西武百貨店9F コンピュータ・フォーラム 03(981)0111	大阪	北区	旭屋書店本店 06(313)1191
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265		都島区	駿々堂京橋店 06(353)2413
	//	有隣堂ルミネ店 045(453)0811	京都	中京区	オーム社書店 075(221)0280
	藤沢	有隣堂藤沢店 0466(26)1411	愛知	名古屋	三省堂名古屋店 052(562)0077
				//	パソコンΣ上前津店 052(251)8334
			刈谷		三洋堂書店刈谷店 0566(24)1134
			長野	飯田	平安堂飯田店 0265(24)4545
			北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ、必要事項を明記のうえ、郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので、大切に保管してください。なお、すでに定期購読をご利用の方には期限終了の

少し前にご通知いたします。継続希望の方は、上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店、日本IPS(株)にお申し込みください。なお、購読料金は郵送方法、地域によって異なりますので、下記宛必ずお問い合わせください。

日本IPS株式会社

〒101 東京都千代田区飯田橋3-11-6

☎03(238)0700



8月号

■1990年8月1日発行 定価560円(本体544円)

■発行人 孫 正義

■編集人 橋本五郎

■発売元 ソフトバンク株式会社

■出版事業部 〒108 東京都港区高輪2-19-13 NS高輪ビル

Oh!X編集部 ☎03(5488)1309

出版営業部 ☎03(5488)1360 FAX 03(5488)1364

広告センター ☎03(297)0181

■印刷 凸版印刷株式会社

©1990 SOFTBANK CORP. 雑誌 02179-8 本誌からの無断転載を禁じます。

落丁・乱丁の場合はお取り替えいたします。



郵便はがき

料金受取人払

高輪局承認

1459

差出有効期間
平成4年7月
15日まで

1 0 8 - 0 0

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!  編集部行

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について	
いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌
推薦する市販ソフト ソフト名： 推薦理由： 1カ月のおこづかいはいくらですか	
あなたの愛機は(所有機種に○印をつけてください) ない X1(マニアタイプ,C,D,F,G,twin) X1turbo(model 10,20,30,40, II,III,Z,ZII,ZIII) MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861) X68000(無印,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER, HD) その他 FD(基) TAPE QD HD(MB) プリンタ()	
年齢	歳
パソコン歴	年
男・女	プレゼントNo.

振替用紙

点線から、きれいに切り取ってご使用ねがいます。

通常払込料金
加入者負担

通常払込料金
加入者負担

払込通知票

通常払込料金
加入者負担

口座番号	東京 1 □ 29307	加入者名	ソフトバンク株式会社	金額	準備	払込人住所氏名	受付局日付印

記載事項を訂正した場合は、その箇所に訂正印を押してください。

口座番号	東京 1 □ 29307	加入者名	ソフトバンク株式会社	金額	準備	払込人住所氏名	受付局日付印

この払込通知票は、機械で使用しますので、下部の線を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。【郵政省】

各票の※印欄は、払込人において記載してください。

フリガナ		性別	年齢	職業
フリガナ		男・女		
住所				
〒				
お電話	自	宅	勤	務
先				
THE COMPUTER	定期購読	新規申し込み	継続申し込み	年間 7,200円
Oh!PC	定期購読	新規申し込み	継続申し込み	年間11,440円 6ヶ月 5,720円
Oh!X	定期購読	新規申し込み	継続申し込み	年間 6,720円
Oh!FM	定期購読	新規申し込み	継続申し込み	年間 6,720円
月刊情報処理試験	定期購読	新規申し込み	継続申し込み	年間 8,160円 6ヶ月 4,080円
C MAGAZINE	定期購読	新規申し込み	継続申し込み	年間11,760円
パソコンマガジン	定期購読	新規申し込み	継続申し込み	年間 6,960円
Beep	定期購読	新規申し込み		年間 5,760円
通付欄				

この払込通知票は、機械で使いますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。 (郵政省)

この欄は、加入者あての通信に使用してください。切り取らないで郵便局にお出しください。

定期購読のお申し込みを頂きありがとうございます。この申込書の弊社到着後切は次の通りです。これを過ぎますと次号からの発送となりますので、ご了承下さい。尚、郵便振替は郵便局で払い込まれてから当社に到着するのに、2週間位かかります。

Oh!PC 1日発売 前月 20日
15日発売 当月 5日

月刊情報処理試験 前月 末日

Beep メガドライブ

Oh! X

Oh! FM

THE COMPUTER 当月 10日

C MAGAZINE

パソコンマガジン

(御注意)

当月発売の号より前に遡ってからの申し込みは出来ません。

ペイマンナンバーは、また別に申し込み下さい。

切り取り線

まるごと新作ボリュームアップ号

POWERFUL MEGA-MAGAZINE

BEEP!

MEGADRIVE

ビーブ!メガドライブ

8月号

定価480円(税込)
好評発売中

創刊1周年記念
ご愛読感謝!

222名プレゼント



〈特別付録〉
ヘルファイアー
特製ポスター

僕達のまわりに異星人がいっぱい
コンピュータゲームの宇宙人侵略史を探る

何かと話題のUFO特集

新作ラッシュの秋を目前に今後の展開を予想する

メガドライブの '90年後半戦を占う

とじこみ保存版

フェリオス攻略ガイド

ヘルファイアー スーパーモナコグランプリ
E SWAT バットマン 四天明王

ゲームボーイ専門誌 パワーアップした第2弾だ!

ゲームボーイLIFE VOL.2

定価380円(税込)

54本のソフトを総ガイド

輝け! 第1回ゲームボーイ大賞

試験にでないゲームボーイ講座

業界初の完全攻略 オールソーサリアンシリーズ

FALCOM MAGAZINE [ファルコム・マガジン]

定価680円(税込)

オールアバウト・ソーサリアン パソコン版ソーサリアン
の総ガイドに加え、メガドライブ版ソーサリアンも紹介
オールファルコム・ベスト10 ファルコムユーザー100人
が選んだファルコムなんでもベスト10

ソフトバンク

ソフトバンクの 書籍特約書店

下記の書店の一覧は、ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。

(注) 現品が売れて補充中の場合もございますので、ご注意ください。

**SOFT
BANK**

ソフトバンク出版事業部

〒108 東京都港区高輪2-19-13 ☎03(5488)1360

全国特約書店一覧

〈北海道〉			浦和市 須原屋コソソ店			048-824-5321			相模原市 文教堂星ヶ丘店			0427-58-6121		
札幌市 紀伊國屋書店札幌店			大宮市 押田謙文堂			048-641-3141			津久井郡 文教堂城山店			0427-82-9278		
〃 旭屋書店札幌店			〃 ブックセンター押田			048-647-3141			〈東京〉					
〃 丸善札幌支店			〃 三省堂ブックポート			048-646-2600			千代田区 三省堂書店神田本店			03-233-3312		
〃 リーブルないわ			茨城県 須原屋蔵店			0484-44-1211			〃 書泉グランデ			03-295-0011		
〃 富貴堂札幌バルコ店			川口市 岩淵書店川口店			0482-52-2190			〃 東京堂書店			03-291-5181		
〃 ダイヤ書房本店			川越市 黒田書店川越店			0492-25-3138			〃 旭屋書店水道橋店			03-294-3781		
〃 ダイヤ書房西店			所沢市 芳林堂所沢店			0429-25-5355			〃 丸善お茶の水店			03-295-5581		
旭川市 旭川富貴堂			〃 いけだ書店所沢店			0429-28-3271			〃 蔵翠堂			03-291-1362		
〃 ブックス平和マルカツ店			上福岡市 黒田書店上福岡店			0492-66-0120			〃 いずみ神田南口店			03-254-8521		
苫小牧市 旭屋書店苫小牧店			朝霞市 文教堂朝霞店			0484-76-0107			〃 明正堂秋葉原店			03-257-0758		
〈東北〉			志木市 新屋堂志木店			0484-74-0182			〃 T-ZONE			03-257-2660		
青森市 成田本店			春日部市 文教堂春日部店			048-752-7666			中央区 八重洲ブックセンター			03-281-1811		
〃 岡田書店			比企郡 錦屋サービス			0492-96-2962			〃 日本橋丸善			03-272-7211		
弘前市 紀伊國屋書店弘前店			千葉市 多田屋セントラルプラザ店			0472-24-1333			〃 旭屋書店銀座店			03-573-4936		
〃 ブックイン城東			〃 キディランド千葉店			0472-25-2011			港区 書原新橋店			03-591-8738		
八戸市 伊吉書院			習志野市 蔵翠堂			0474-72-5011			〃 雄略堂N.S.店			03-503-6586		
盛岡市 東山堂書店本店			船橋市 ときわ書房本店			0474-24-0750			〃 虎ノ門書房本店			03-502-3461		
〃 さわや書店			〃 リプロ船橋店			0474-25-0111			〃 虎ノ門書房田町店			03-454-2571		
〃 第一書店			〃 旭屋書店船橋店			0474-24-7331			品川区 芳林堂大井町店			03-474-4946		
仙台市 金港堂			〃 芳林堂津田沼店			0474-78-3737			〃 明屋書店五反田店			03-492-3881		
〃 金港堂ブックセンター			〃 第二蔵翠堂			0474-65-0926			渋谷区 紀伊國屋書店渋谷店			03-463-3241		
〃 アイエ書店駅前店			〃 三省堂書店西船橋店			0474-34-3111			〃 旭屋書店渋谷店			03-476-3971		
〃 丸善仙台支店			柏市 西アサノ			0471-44-2111			〃 三省堂書店渋谷店			03-407-4545		
〃 高山書店			〃 新屋堂柏店			0471-64-8551			〃 大盛堂書店			03-463-0511		
〃 ブックスみやぎ			松戸市 丸屋良文堂			0473-65-5121			〃 紀伊國屋書店笹塚店			03-485-0131		
秋田市 三浦書店			〃 辰正堂駅ビル店			0473-64-7997			新宿区 紀伊國屋書店本店			03-354-0131		
山形市 八文字屋			横浜市 有隣堂トーヨー店			045-311-6265			〃 三省堂書店新宿西口店			03-343-4871		
福島市 岩瀬書店コルニエツタヤ店			〃 有隣堂東口ルミネ店			045-453-0811			〃 福家書店センタービル店			03-345-1246		
〃 博尚堂			〃 栄松堂相鉄ジョイナス店			045-321-6831			〃 福家書店野村ビル店			03-342-0298		
郡山市 東北書店			〃 そごうブックセンター			045-465-2111			〃 新屋堂N.S.ビル店			03-344-2055		
いわき市 ヤマニ書房本店			〃 丸善ブックメイツポルタ店			045-453-6811			〃 西武新宿ブックセンター			03-208-0380		
〃 鹿島ブックセンター			〃 有隣堂伊勢佐木店			045-261-1231			〃 芳林堂高田馬場店			03-208-0241		
会津若松市 宝文館			〃 有隣堂戸塚店			045-881-2661			〃 未来堂			03-200-9185		
原町市 文芸堂			〃 文華堂戸塚店			045-864-5151			豊島区 旭屋書店池袋店			03-986-0311		
〈関東〉			〃 アーバン文華堂			045-821-5151			〃 芳林堂池袋店			03-984-1101		
水戸市 川又書店駅前店			〃 文芸堂青葉台南口店			045-983-5150			〃 リプロ池袋店			03-981-0111		
〃 ツリヤブックセンター			川崎市 有隣堂アゼリア店			044-245-1231			〃 三省堂書店池袋店			03-987-0511		
勝田市 武石書店			〃 有隣堂川崎BE店			044-200-6831			〃 新栄堂本店			03-984-2345		
東海村 大野書店			〃 文学堂本店			044-244-1251			〃 新栄堂アルパ店			03-988-0181		
鹿島郡 なみき書店			〃 文芸堂溝ノ口店			044-811-8258			台東区 明正堂中通り店			03-831-0191		
土浦市 共栄堂			鎌倉市 鎌倉書店大船店			0467-46-3841			墨田区 ブックストア・談			03-635-1841		
つくば市 丸善筑波大学会館店			〃 鎌倉書店			0467-46-2619			葛飾区 文教堂青戸店			03-838-5938		
〃 友朋堂吾妻本店			横須賀市 平坂書房WALK店			0468-25-5537			江戸川区 文教堂西葛西店			03-689-3621		
宇都宮市 落合書店オリオン店			藤沢市 有隣堂藤沢店			0466-26-1411			大田区 アクトブックスサンカマタ店			03-735-1551		
〃 落合書店東武ブックセンター			〃 リプロ藤沢店			0466-27-0111			〃 竜文堂大森駅ビル店			03-775-3851		
〃 新屋堂宇都宮店			茅ヶ崎市 文芸堂六会店			0466-82-9610			中野区 明屋書店東京本社			03-387-8451		
小山市 進徳堂駅ビル店			〃 川上書店ルミネ店			0467-87-3827			杉並区 ブックセンター荻窪			03-393-5571		
前橋市 煥乎堂			平塚市 サクラ書店駅ビル店			0463-23-2751			〃 書原杉並店			03-313-4778		
〃 リプロ前橋店			小田原市 文教堂四之宮店			0463-54-2880			武蔵野市 紀伊國屋書店吉祥寺東急店			0422-21-5543		
〃 戸田書店前橋店			〃 八文書店			0465-22-7111			〃 弘栄堂吉祥寺店			0422-22-1031		
高崎市 学陽書房			〃 伊勢治書店			0465-22-1366			〃 バルコブックセンター吉祥寺			0422-21-8122		
〃 サカサ書店			〃 文教堂小田原店			0465-36-3677			調布市 真光書店			0424-87-2222		
〃 新屋堂高崎店			〃 有隣堂厚木店			0462-23-4111			府中市 啓文堂			0423-66-3151		
〃 戸田書店高崎店			大和市 文教堂中央林間店			0462-75-4165			三鷹市 三省堂書店三鷹店			0422-48-4510		
太田市 ナカムラヤ			相模原市 文教堂相模大野店			0427-49-0650			〃 東西書房			0422-46-0275		
〈首都圏〉			〃 文教堂橋本店			0427-74-5581			小金井市 文教堂小金井店			0423-86-0161		
浦和市 須原屋本店									国分寺市 三成堂国分寺店			0423-25-3211		

展示図書一覧

定価は本体価格です。

MS-DOSいたれりつくせり本 ●1800円
 プレイMS-DOS ●1900円
 UNIX System V
 プログラマ・ガイド ●12000円
 UNIX System V
 ユーザ・ガイド ●9800円
 UNIXオペレーティングガイド ●3000円
 OS/2 APIブックI ●2709円
 C言語の活用理解 ●2000円
 C言語の基礎知識 ●2500円
 C言語の応用50例 ●2300円
 上級・C言語の応用例50例 ●2400円
 Cプリプロセッサ・パワー ●2200円
 Play the C 上・下 ●各1500円
 Turbo C入門 ●2600円
 C++プログラミング ●2600円
 Quick Cプログラミング ●2602円
 詳説C言語 ●4369円
 8086アセンブリ言語 ●2800円
 8086マクロプログラミング ●2600円
 Final Ver.4.0ブック ●2400円

MIFES Ver.4.0ブック ●2400円
 ビジネスソフトデータ活用ブック ●2800円
 BASICによるプログラミング
 スタイルブック ●1800円
 ソーティング・ノート ●1900円
 J-3100パワーユーザーブック ●2400円
 続・PC工作入門 ●1800円
 PC-286Lブック ●1700円
 試験に出るX1 ●2800円
 RDBファラオ活用ガイド ●2903円
 言図ガイド ●2301円
 Rydeenガイド ●2427円
 P1 EXEガイド ●2524円
 Lotus1-2-3ガイドII ●2500円
 MS-Chart Ver.3.1ガイド ●2900円
 まいと〜くガイド ●2300円
 新松ガイド ●2000円
 一太郎Ver.3ガイド ●2500円
 新一太郎ガイド ●2300円
 桐Ver.2ガイド ●2500円
 花子応用ガイド ●2500円

Lotus 1-2-3ガイド ●2400円
 P1ガイド ●2300円
 Ninja2 ガイド ●2300円
 Multiplan
 Ver.3.1ガイド ●2400円
 アセンブラCASL入門 ●2000円
 ハードウェア徹底マスター ●2500円
 FORTRAN徹底マスター ●2800円
 情報処理の基礎知識 ●1600円
 COBOL 徹底マスター ●2900円
 受験用語ハンドブック ●1800円
 情報処理入門1・2 ●各1204円
 CASLで学ぶ
 アセンブラ言語入門 ●2204円
 バイト&ワードの風について ●1800円
 田原総一郎のパソコンウォーズ ●1400円
 パソコンを襲う
 知的独占の戦い ●1600円
 RPG幻想事典・日本編 ●1800円
 魔法王国シムルグント ●1800円

国立市 東西書店 0425-75-5061
 小平市 文教堂小平店 0423-43-9229
 東村山市 文教堂東村山店 0423-96-1115
 立川市 オリオン書房ウエル店 0425-27-2311
 八王子市 くまざわ書店本店 0426-25-1201
 町田市 有隣堂町田店 0427-23-3018
 久美堂本店 0427-25-1330
 久美堂小田急店 0427-27-1111
 文教堂鶴川店 0427-35-4117
 文教堂小川店 0427-96-1781
 多摩市 くまざわ書店桜ヶ丘店 0423-37-2531
 福生市 文教堂福生店 0425-53-7708
 <甲信越・北陸>
 甲府市 文教堂甲府店 0552-22-4600
 長野市 平安堂長野店 0262-26-4545
 長谷川書店 0262-26-2122
 上田市 平安堂上田店 0268-22-4545
 松本市 ブックスロクサン 0263-35-5555
 改進黨松本駅ビル店 0263-36-3777
 飯田市 平安堂飯田店 0265-24-4545
 岡谷市 並原書店 0266-23-5070
 諏訪市 平安堂下諏訪店 0266-28-1111
 新潟市 紀伊國屋書店新潟店 025-241-5281
 萬松堂 025-229-2221
 北光社 025-228-2321
 長岡市 寛張書店 0258-32-1139
 ブックセンター長岡 0258-36-1360
 長岡技大長峰文化 0258-46-6437
 パソピア コスモス 0255-25-5867
 BOOKメディア 0254-77-3850
 富山市 瀬川書店 0764-24-4566
 清明堂 0764-24-4166
 BOOKSなかた豊田店 0764-32-1353
 文苑堂本郷店 0764-22-0552
 文苑堂赤江店 0764-33-0321
 高岡市 文苑堂 0766-21-0333
 文苑堂横田店 0766-21-0431
 金沢市 うつのみや片町店 0762-21-6136
 書林香林坊本店 0762-20-5011
 野々市 王様の本本店 0762-46-5325
 福井市 勝本書店 0776-24-0428
 品川書店新田塚店 0776-24-1112
 <東海>
 静岡市 静岡谷島屋呉服町本店 0542-54-1301
 江崎書店 0542-54-4481
 吉見書店 0542-52-0157
 戸田書店SBS店 0542-81-5733
 戸田書店曲金店 0542-81-5899
 沼津市 吉野屋 0559-23-5676
 マルサン書店宝塚店 0559-63-0350
 富士市 戸田書店富士店 0545-51-5121
 清水市 戸田書店本店 0543-65-2345
 浜松市 浜松谷島屋連尺本店 0534-53-9121
 名古屋 三省堂書店名古屋店 052-562-0077
 星野書店近鉄ビル店 052-581-4796
 丸善名古屋支店 052-616-2251
 丸善ブックメイソントラルパーク 052-971-1231
 日進堂上前津店 052-263-0550

名古屋 三洋堂パソコンショップ 052-251-8334
 三洋堂いりな本店 052-832-8202
 ちくさ正文館本店 052-741-1137
 白樺書房西店 052-774-7223
 豊橋市 精文館 0532-54-2345
 岡崎市 ブックス鎌倉 0564-54-1822
 豊田 三洋堂梅坪店 0565-35-2334
 豊川市 三洋堂豊川店 05338-3-0334
 刈谷市 三洋堂刈谷店 0566-24-1134
 春日井市 三洋堂勝川店 0568-32-7806
 岐阜市 自由書房 0582-65-4301
 大垣市 大洞堂ブックス258 0584-81-2553
 大洞堂岐大バypass店 0584-74-7766
 一宮市 三洋堂一宮店 0586-77-5734
 多治見市 三洋堂多治見店 0574-63-2334
 津市 別所書店11ビル店 0592-24-1014
 四日市市 文化センター白揚 0593-51-0711
 鈴鹿市 シェルト白揚ススカ 0593-82-5221
 <近畿>
 京都市 駿々堂京京店 075-223-1003
 アンパティ・ブックセンター 075-682-5031
 オーム社書店河原町店 075-221-0280
 ジュンク堂京都店 075-252-0101
 オーム社書店竹田店 075-644-2611
 奈良市 駿々堂大丸店 0742-26-6241
 大阪市 旭屋書店本店 06-313-1191
 紀伊國屋書店梅田店 06-372-5821
 オーム社書店大阪店 06-345-0641
 駿々堂京橋店 06-353-3209
 駿々堂心斎橋店 06-251-0881
 旭屋書店ナンバ店 06-644-2551
 ナンパブックセンター 06-644-5501
 旭屋書店アベノ店 06-631-6051
 ユーゴー書店 06-623-2341
 河村書店 06-951-2968
 枚方市 水嶋書房京阪デパート店 0720-51-3432
 高槻市 コーベックス西武高槻店 0726-83-1766
 東大阪市 ヒバリヤ書店本社 06-722-1121
 神戸市 ジュンク堂センター街店 078-392-1001
 ジュンク堂サンバル店 078-252-0777
 海文堂書店 078-331-6501
 日東館書林 078-391-8701
 姫路市 新誠書房 0792-85-3344
 新誠堂書店 0792-81-2055
 和歌山市 宮井平安堂 0734-31-1331
 帯伊書店 0734-22-0441
 <中国>
 岡山市 紀伊國屋書店岡山店 0862-32-3411
 丸善岡山支店 0862-31-2261
 津山市 津山ブックセンター 08682-6-4047
 広島市 紀伊國屋書店広島店 082-225-3232
 丸善広島支店 082-247-2251
 金正堂 082-248-3715
 横書館 082-248-3151
 尾道市 啓文社尾道店 0848-37-5151
 福山市 啓文社福山店 0849-22-3111

福山市 ブックシティ啓文社 0849-25-0050
 啓文社コア 0849-41-0909
 山口市 五十部誠文堂 0839-24-6630
 文栄堂 0839-22-5611
 下関市 中野書店 0832-22-6181
 宇部市 京屋書店 0836-31-2323
 末広書店 0836-31-0086
 防府市 誠文堂国術店 0835-25-1988
 光市 三文字屋 0833-71-0251
 鳥取市 富士書店 0857-23-7271
 松江市 園山書店 0852-21-4167
 <四国>
 徳島市 小山助学館本店 0886-54-2135
 小山助学館東口店 0886-25-1380
 森住九善 0886-23-3228
 高松市 宮脇書店本店 0878-51-3733
 丸亀市 宮脇書店丸亀店 0877-22-5533
 松山市 紀伊國屋書店松山店 0899-32-0005
 明屋書店本店 0899-41-4141
 明屋書店大街道店 0899-41-4242
 九三書店 0899-31-8501
 新居浜市 明屋星原店 0897-44-4000
 宇和島市 明屋宇和島店 0895-23-1118
 高知市 金高堂 0888-22-0161
 <九州・沖縄>
 福岡市 紀伊國屋書店福岡店 092-721-7751
 リーぷる天神 092-713-1001
 積文館新天町店 092-781-2991
 福岡金文堂本店 092-741-2106
 福岡金文堂朝日ビル店 092-431-1094
 福岡金文堂デイトス店 092-451-6175
 福岡金文堂アニマト原 092-844-0088
 ナガリ書店 093-521-1044
 金栄堂 093-531-3685
 旭屋書店北九州店 093-631-6421
 井筒屋ブックセンター 093-641-0131
 カルバーク平野 093-661-7988
 白石書店本城店 093-601-2200
 エマックスたがみ 0942-33-1841
 飯塚市 BOOKリード 0948-25-0266
 大分市 バルコブックセンター大分店 0975-35-7643
 本町昇屋 0975-33-0231
 別府市 明林堂 0977-23-2183
 宮崎 中央、田中書店 0985-24-3151
 寿屋宮崎店 0985-27-4111
 佐賀市 金華堂北バイパス店 0952-32-1965
 横文館佐賀店 0952-24-4314
 横文館デイトス店 0952-23-7155
 長崎市 メトロ書店 0958-21-5453
 好文堂 0958-23-7171
 佐世保市 金明堂書店 0956-22-4214
 熊本市 紀伊國屋書店熊本店 096-322-5531
 長崎書店 096-353-0555
 人吉市 明屋人吉店 0966-22-5486
 鹿児島市 春苑堂ブックプラザ 0992-25-3200
 ブックスすみ 0992-57-1011
 那覇市 球陽堂書房ビル店 0988-63-3752
 文教図書 0988-62-1201

業界研究セナ一

199

LANE
株式会社

日本最大規模の
LANソフト専門会社、
社を発足させた。変貌す。

EEK

[illegible]

集合研修

く、また「海外旅行
たい所」はヨーロッパ
アメリカ、オセアニアの
あった。
フレッシュな息吹を会

日本ソフトバンク
社名を
「ソフトバンク
と改称」

雀日
ンク株式

開催日決定！
 ソフトバンク株式会社セミナー
 電界講習！
 一たがム

会場

札 幌：7月23日
仙 台：7月20日
東 京：7月16、24日
名古屋：7月26日
大 阪：7月 9日
広 島：7月11日

TEL 011 (222) 6026
TEL 022 (263) 0907
TEL 03 (5488) 1115
TEL 052 (261) 7215
TEL 06 (264) 1471
TEL 082 (223) 1314

札仙本名西店
幌台社古屋日本島
營業所營業所人事部營業所營業所營業所

ソフトバンク

〒108 東京都港区高輪2-19-13
NS高輪ビル

TEL 03 (5488) 1115

HOST

△ 68000 専用
多回線 ホストソフト

PRO-68K

ついに
登場!

3回線 / 9回線

きみも、今日から局長さん

HOST 9 PRO-68K 概要

対応回線数 1~9回線
使用モデム ATモデム MNP(RTS/CTS)可
通信速度 最大9600bps
会員数 *最大9999人
掲示板数 *最大40個
機能 電子掲示板・電子手帳・電子会議(チャット)・会員情報
これらは、コンフィグファイルで設定できます。

注1: *印について拡張を希望する場合は、プログラムの書き換えが必要になりますので、別料金にて対応致します。当社までご相談ください。
2: 2回線以上で運用される場合は、CZ-68F1(シャープ純正)が必要になります。
3: このホストはテキスト形式の転送方法を採用しております。

■特長

- 各種設定のコンフィグファイル化。●RS-232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能。●モニター、各チャンネルのユーザーの打ち込んだコマンドや通信状態を確認。●各掲示板別にSIG、ポートパスの設定。●メンテナンス作業のオンライン実行。(ボードインテックス、メールインテックス)●オンラインサインアップ等、ゲストへの設定が可能。●通信サービスTri-P対応。●行編集(オンライン簡易エディタ)機能。●その他、シスオペレベルで会員情報の変更が可能。タイムアウトによる回線切断。PDS専用掲示板の採用。(1書込中で、ドキュメントとテキストプログラムの分離)。●接続MNPタイプの識別。●ログイン、ログアウト時間の記録。●非アクセス時のモニター画面消去可能。

HOST 3 PRO-68K

機能はすべて、「HOST 9 PRO-68K」と同じですが、対応回線数が、1~3回線に制限されて、低価格でユーザーに供給します。

バージョンアップ (Ver1.10) サービス実施中

現在発売されています製品は、Ver1.10に変更になっています。お使いの製品がVer1.00のユーザーの方のために、バージョンアップサービスを実施しておりますので、お早目に、ユーザー登録葉書をお送り下さい。

Ver1.10へ無料交換を実施しております。

好評発売中

HOST 9 PRO-68K ¥59,800円

HOST 3 PRO-68K ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

今、X68000の
通信が変わる!!

好評発売中
17,800円

24/31KHz
ディスプレイ
対応

△ 68000 専用
パソコン通信ソフト

た〜みのる

2

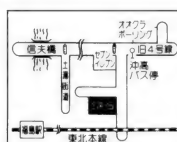
「た〜みのる」が
装いも新たに
「た〜みのる2」として登場!
「た〜みのる」が
通信人門版なら
「た〜みのる2」は
マニアタイプの通信ソフトです!!

「た〜みのる2」はX68000用に製作された通信ソフトです。
X68000の機能を充分に引き出して、ユーザーのみなさんが簡単に
に操作できるように工夫・製作されています。

プログラマ募集!!

SPSでゲームを作ってみませんか?

アセンブラでプログラムの組める優秀な人材を若干名募集しています。就職希望の方は62円切手同封の上、「就職案内係 大和」までお手紙ください。折り返し就職のご案内をお送り致します。尚、デザイナー、音楽プログラム等の専門職は募集しておりません。



(株)マイコンハウス

SPS

〒960 福島市太平寺字町/内5-3 ☎(0245)45-5777
FAX(0245)45-1804(G11, G111)

当社の製品は全国の有名デパート、パソコンショップでお求めになります。尚、お求めにない場合、郵便局にてお申し込みください。●口座番号 郡山5-12298
●加入者名 関エス・ピー・エス ●金額 代金に3%の消費税を計算した額 ●通信費 (裏面) ●希望ゲームソフト名、数量、代金合計、年齢、氏名、機種名、テープかディスクの種類。(一週間以上かかりますので、お急ぎの方は現金書留をご利用ください。その場合、おつりのいらないうにお願いします。

■表示価格に消費税は含まれておりません。

△ 68000
HOST PRO-68K 使用

SPS-NET

TEL
(0245)46-1167代

好評 一般回線
運営中 (9回線)
(4回線) MNPクラス7

24時間運営 (N81XN)
ゲストID (GUEST)

*GUESTアクセスは無料です。ぜひ、一度試してください。

例◎パスワード=SPS-NET
(8文字まで大文字の識別あり)

◎本名=大和五郎(8文字まで)

◎ペンネーム=大ちゃん(4文字まで)

◎年齢=30(現在の年齢)

◎電話=0245-45-5777(市外局番から)

入会方法 登録料¥3,000(税別)
会費無料

下記の用紙に直接記入するか又は、コピーして記入し、72円切手同封の上、「SPS-NET係」までお送り下さい。届き次第、仮登録を行いID発行後SPS-NET専用の郵便振込み用紙ならびに運用の手引きをお送りいたします。それに従い、3ヶ月以内に登録料3,000円(税別)を御入金下さい。入金確認後正式会員として再登録します。

◎職業=株式会社エス・ピー・エス(16文字まで)

◎住所=福島市太平寺字町/内5-3(24文字まで)

◎自己紹介=SPS-NETをよろしく(24文字まで)

◎システム構成=X68000ACE-HD MD2400B(18文字まで)



専用

turbo OK-システム 漢字

「個人簿記会計 財計くん」2HD版
定価 49,800円 (税別)

出力帳票：勘定科目一覧表・摘要一覧表・期首貸借対照表・期末試算表・貸借対照表・損益計算書・仕訳帳・各科目別元帳・合計残高試算表

処理金額 9桁10億円/年間
月間仕訳処理数 900件以内
仕訳入力は一度 振替伝票方式採用
使用勘定科目数 75個(年度変更可)
摘要小書き入力 A・Bの2つ
Aはコード入力
Bは自由入力
オート・ソート 仕訳訂正で
日付自動処理
ラクラク金額入力 カンマ付き、無
どちらもOK!

消費税の会計処理 注目の消費税の
会計処理は、4つの対応が考えられ
ますが、ユーザー別に勘定科目の設
定をする事により処理できます。

「消費税検証」を別冊にて同梱し
てあります。ご活用下さい。
〈各種税法は変化しても、複式簿記
の原理は不変です。勘定科目の設定
によって処理できるのが、財計くん
なのです。〉

プリンター用紙

縦11イン치의白紙又は罫線入りを
使用願います。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システムユーザー辞書使用可。
2. 科目&摘要の入力時にHTLPキー機能を追加。

「個人簿記会計 財計くん」2D版
定価 39,800円 (税別)

2HD版との相違は、先の能力アップの内容の通りです。

各資料のご請求は

資料は、一部あたり200円分の切手を同封願います。各デモ・サンプル版は実費2400円を申し受けます。

弊社へ直接お申込みの方は上記分を差し引いてご本体を購入できます。

資料は毎月曜日に、デモ版は逐次発送しています。

「財計くん 売掛管理台帳」2HD版
定価 39,000円 (税別)

出力帳票：納品書・請求書・アイウエオ順顧客一覧表・取扱商品一覧表・売上日計表・売掛残高一覧表・DMシール (条件検索可)

処理金額 9桁10億円/年間
1顧客処理件数 60件/月間 繰越可
処理顧客数 1DataDisk 1200名
取扱商品数 1DataDisk 250品目
消費税自在処理 登録済使用と未登録
使用どちらも可
登録済顧客変更 台帳変更Bで自在
帳票3段階選択 顧客別&メ切&全部
商品単価無登録 250品目が無限に
ラクラク金額入力 カンマ付き、無
どちらもOK!

プリンター対応表

ご使用になる機種により4つのシリーズ品番がございます。ご購入の際にはご確認願います。

No701:CZ-8PK3・CZ-8PK4・CZ-8PK5・C-8PK6・CZ-8PK7・CZ-8PK8・CZ-8PK9・EPSO N-VPシリーズ=X1ROM要
No702:CZ-8PK2・CZ-80PK
No703:CZ-8PD2・CZ-8PD2・CZ-800P・EPSON-SPシリーズ=X1ROM要

No704:X1に接続可能なもので、縦11イン치의白紙又は罫線入りのもののみを利用する事になります。

* 伝票専用紙として、ヒサゴ(株)GB-342を使用します。伝票以外は縦11イン치의連続用紙(白紙or罫線入り)を使います。なお、No.704のみは、伝票用紙はユーザーが作成して使用する事になります。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2. 商品名の入力時にHELPキー機能が追加。

「財計くん 売掛管理台帳」2D版
定価 29,000円 (税別)

2HD版との相違は、先の能力アップの内容の通りと、処理顧客数が600名となり、取扱商品数が150品目となります。(2HD同様No701~No.704品番がございます。ご購入の際はご確認下さい。)

「DATA・CARD 1200」2HD版
定価 42,000円 (税別)

カード型データベースとしての機能とグラフ作成ツールのグラフデーター・ファイル機能を持っています。検索は、1,124枚のデーターカードから3重条件を処理します。

項目設定は自由設定で12個までを処理し、データー部は新規に設けました「データー変換Uty」で、作成済みのデーターでもデーター量に応じて変更可能になりました。

DMシール発行・葉書宛名印刷を条件検索で処理します。

カードNoによる、データーの抜粋・ステップ印刷(同カードを最大12枚まで)を処理します。

グラフ・ツールとしては、7種・22タイプのグラフを作成する事ができ、最大12項目12データーを縦棒グラフ・横棒グラフ・帯グラフ・円グラフ・折線グラフに処理します。縦棒グラフ・横棒グラフは3D仕様でも処理します。

プリンター用紙

縦11イン치의白紙又は罫線入りを使用願います。

2D版との能力アップの内容

1. ディスクの入れ替えなしで、システム・ユーザー辞書使用可。
2. グラフDataDisk内に格納できるファイル数が3倍になりました。

「DATA・CARD 1200」2D版
定価 32,000円 (税別)

2HD版との相違は、先の能力アップの内容の通りです。

ご購入は

お近くのパソコン・ショップでお求め下さい。お急ぎの方は直接現金書留でお申し込み下さい。

(売掛管理台帳のNo704のみユーザーのご希望により、プログラム解放型2D¥58,000円(税別)もあります。直接弊社にお申し込みください。)

〒885 宮崎県都城市都島町430-2

OK-ハウス

TEL 0986-25-0303-FAX 0986-25-9553

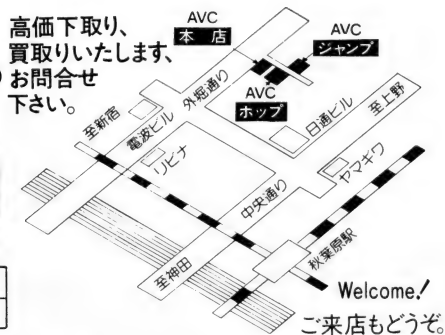


AVCフタバ

03(253)7661

〒101 東京都千代田区外神田3-2-3 ☎03-253-7611(代)

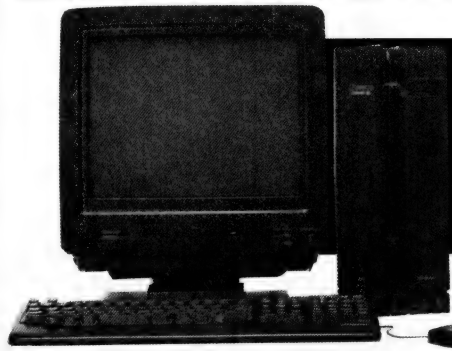
今すぐ もよりの電話から	仙台 022-264-3704	名古屋 052-452-3271	広島 082-295-6873
札幌 011-611-5104	新潟 0252-75-4175	大阪 06-311-3931	福岡 092-481-2494



X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

68000 待望の新しい仲間登場!!

PERSONAL WORKSTATION
EXPERT II・EXPERT II HD



EXPERT II・EXPERT II HD
集積度を高めたマンハッタンシェイプ™3Mの大容量メモリを搭載。本格的なウィンドウシステム、SX-WINDOW搭載。

(写真のモニターは別売です。)

CZ-603C 標準価格 ¥338,000
CZ-613C 標準価格 ¥448,000

AVC 特価

68000

PERSONAL WORKSTATION
PRO II・PRO II HD



PRO II・PRO II HD
拡張I/Oポートを4スロットを搭載し、汎用性と低価格が魅力。もちろん、SX-WINDOW搭載。

CZ-653C 標準価格 ¥285,000
CZ-663C 標準価格 ¥395,000

AVC 特価

X68000		お勧めディスプレイコーナー 組合せは自由、価格はお気軽にご相談下さい。			
 在庫稀少価格はお電話で!	CZ-602C	CZ-604D 標準価格 ¥94,800 AVC 特価	●0.31mmドットピッチ ●2モードオートスキャン ●ステレオスピーカー搭載 ●チルト台同梱	CZ-613D 標準価格 ¥135,000 AVC 特価	●ドットピッチ 0.31mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱
	CZ-612C	CU-21HD 標準価格 ¥148,000 AVC 特価	●0.52mmドットピッチ ●21型ディスプレイ ●3モードオートスキャン ●ステレオスピーカー搭載	CZ-605D 標準価格 ¥115,000 AVC 特価	●ドットピッチ 0.39mm ●TVチューナー搭載 ●ステレオスピーカー搭載 ●チルト台同梱
	CZ-652C				
	CZ-662C				

型番	品名	標準価格	販売価格	型番	品名	標準価格	販売価格	型番	品名	標準価格	販売価格
CZ-6TU	システムチューナー	¥ 33,100	AVCフタバ特価	CZ-8PG1	24ピンカラープリンター (80桁)	¥ 130,000	AVCフタバ特価	CZ-8TM2	モデムユニット	¥ 49,800	AVCフタバ特価
BF-68PRO	CRTフィルター	¥ 19,800	AVCフタバ特価	CZ-8PK10	24ピンプリンター (136桁)	¥ 97,800	AVCフタバ特価	CZ-252MS	MusiStudio	¥ 28,800	AVCフタバ特価
CZ-8NS1	カラーレスキャナー	¥ 188,000	AVCフタバ特価	IO-735X	カラージェットプリンター	¥ 248,000	AVCフタバ特価	CZ-247MS	MUSIC (MID)	¥ 28,800	AVCフタバ特価
CZ-6BN1	スキャナー用パラレルボード	¥ 29,800	AVCフタバ特価	CZ-6BE1A	1M増設 RAMボード	¥ 38,000	AVCフタバ特価	CZ-221HS	NEW Print Shop	¥ 19,800	AVCフタバ特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	AVCフタバ特価	CZ-6BE2	2M増設 RAMボード	¥ 79,800	AVCフタバ特価	CZ-228BS	TOP給与計算エクスパート	¥ 200,000	AVCフタバ特価
CZ-8BV2	カラーイメージボード	¥ 39,800	AVCフタバ特価	CZ-6BE4	4M増設 RAMボード	¥ 138,000	AVCフタバ特価	CZ-227BS	TOP財務会計	¥ 200,000	AVCフタバ特価
CZ-8BR1	立体映像セット	¥ 29,800	AVCフタバ特価	CZ-6BP1	数値演算プロセッサ	¥ 79,800	AVCフタバ特価	CZ-220BS	DATA	¥ 58,000	AVCフタバ特価
CZ-8DT2	パーソナルテロップ	¥ 44,800	AVCフタバ特価	CZ-6BC1	FAXボード	¥ 79,800	AVCフタバ特価	CZ-212BS	BUSINESS	¥ 68,000	AVCフタバ特価
CZ-8BS1	FM音源ボード	¥ 23,800	AVCフタバ特価	CZ-6BM1	MIDIボード	¥ 26,800	AVCフタバ特価	OS-9		¥ 29,800	AVCフタバ特価
CZ-8NJ1	ジョイカード	¥ 1,700	AVCフタバ特価	CZ-6BU1	I/Oボード	¥ 39,800	AVCフタバ特価	CZ-211LS	Compiler	¥ 39,800	AVCフタバ特価
CZ-8NM2A	マウス	¥ 6,800	AVCフタバ特価	CZ-6BL1	LANボード	¥ 268,000	AVCフタバ特価	CZ-234LS	A1-68K	¥ 188,000	AVCフタバ特価
CZ-8NM3	マウス・トラックボール	¥ 9,800	AVCフタバ特価	CZ-243BS	サイバーノート	¥ 19,800	AVCフタバ特価	CZ-620H	20MBハードディスク	¥ 178,000	AVCフタバ特価
CZ-6SD1	システムラック	¥ 44,800	AVCフタバ特価	CZ-240BS	ステーションリ	¥ 14,800	AVCフタバ特価	CZ-64H	40MBハードディスク	¥ 120,000	AVCフタバ特価
AN-S100	アンプ内蔵スピーカー	¥ 36,600	AVCフタバ特価	CZ-223CS	通信ソフト	¥ 19,800	AVCフタバ特価	LHD-34V	40MBハードディスク(ロジック)	¥ 153,000	¥ 117,000
CZ-6EB1	拡張I/Oボックス	¥ 88,000	AVCフタバ特価		ゲームソフト	20% OFF		LHD-32V	20MBハードディスク(ロジック)	¥ 128,000	¥ 98,000



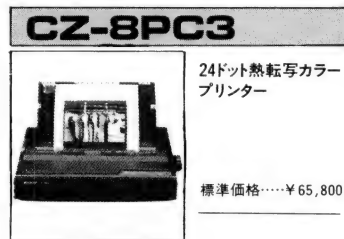
AVC 特価 ¥ ???



AVC 特価 ¥ ???



AVC 特価 ¥64,800



AVC 特価 ¥39,800

●頭金なし(手軽な電話クレジット) ●製品先取り(お支払いは約1~2ヶ月後から) ●低金利クレジット(1回の支払いは2,700円以上で3~48回。ボーナス併用可) ●クレジットクレジット(保証人なし。但し満20歳以上の学生の方) ●18歳未満の方(ご両親が代理購入者としてお申し込み下さい) ●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい) ●完全保証(すべてメーカー保証書付。アフターケア万全) ●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM7時
まで受付 日曜・祝日も営業

●セットの組合せは自由、広きに出ている他の機種はお問合せ下さい。

信用と実績を誇る

BASIC HOUSE

BASIC HOUSE 大田原店

一周年

大謝恩セール!

〈主な取扱いメーカー〉

Apple Computer

FUJITSU

NEC

EPSON

SHARP

TOSHIBA

全品大特価!!

7/20(金)~7/23(月)

BASIC HOUSE 宇都宮

店内改装のため

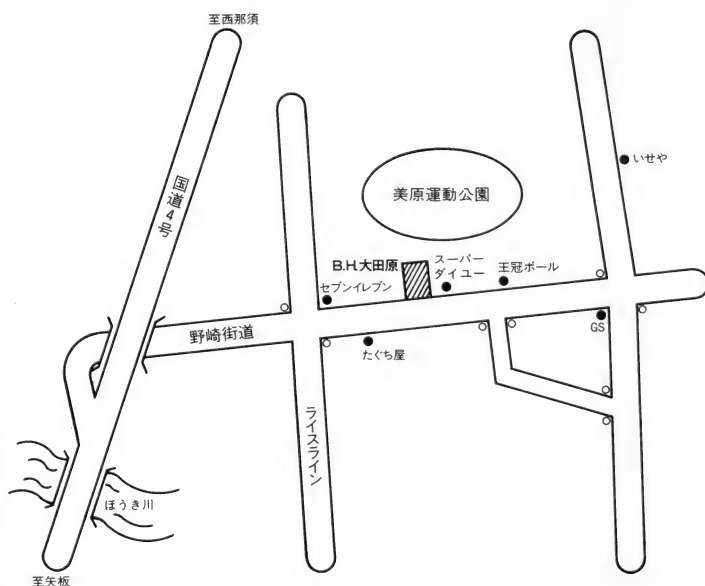
★閉店売りつくしセール

7/20(金)~7/23(月)

★新装開店セール

8/3(金)~8/6(月)

※通信販売もOKです!!



全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部
大田原営業所/マイコンショップ

宇都宮市竹林町503-1 TEL 0286-22-9811 FAX 0286-25-3970
大田原市美原1-13-4 TEL 0287-23-5352 FAX 0286-23-5364

マイコンショップ

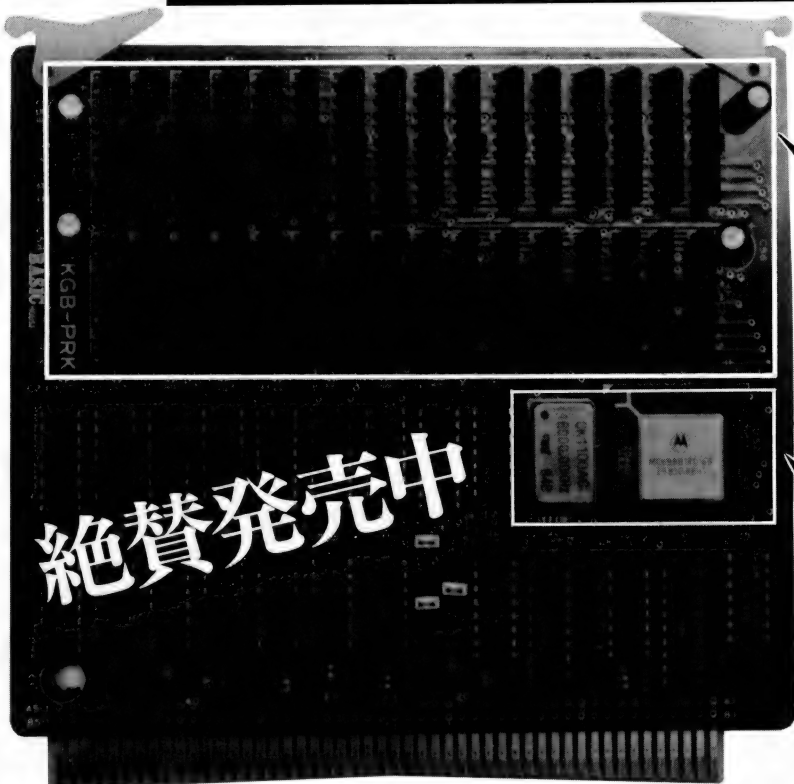
BASIC HOUSE

お申し込み・お問い合わせは

0286-22-9811(代)

2枚のボードが1枚になった

KGB-X68PRK



広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
**数値演算
プロセッサ**

- メモリアクセスノーズウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目/未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※写真はKGB-X68PRK-14です

※CZ-602C/CZ-612C以外の機種ではCZ-6BE1/CZ-6BE1Aを実装している必要があります
※メモリアクセスノーズウェイトのため拡張I/O BOXでは動作しません

製品価格一覧

KGB-X68PRK-01 ￥58,000 (1Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-11 ￥96,000 (1Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-02 ￥74,000 (2Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-12 ￥112,000 (2Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-03 ￥98,000 (3Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-13 ￥136,000 (3Mメモリ/数値演算プロセッサ付き)
KGB-X68PRK-04 ￥122,000 (4Mメモリ/数値演算プロセッサ無し)	KGB-X68PRK-14 ￥160,000 (4Mメモリ/数値演算プロセッサ付き)

購入後の増設費用

メモリ	
1Mバイト	￥24,000
2Mバイト	￥51,000
3Mバイト	￥76,000
数値演算プロセッサ	
MC68881RC16	￥38,000

充実のBASICHOUSEハードウェア&ソフトウェア

高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	￥118,000	高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	￥98,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	￥42,000	汎用ローコストA/D&PIOボード(KGB-X1S) X1	￥19,800
ハードディスクインターフェースボード(KGB-HDIF) X1	￥16,000	高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	￥128,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	￥68,000	64180CPUボードMach180(KGB-CPXB) X68000	￥98,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	￥24,800	ローコストMIDIインターフェース(MELODY BOX) X68000	￥16,800
BASIC拡張関数パッケージ(B6-6301) ￥9,800	C言語ライブラリ(B6-6305) ￥6,800	BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	￥14,800
ディスクキャッシュ(B6-6304) ￥6,800	Toys & Tools (B6-6307) ￥6,800	アイコンエディタ(B6-6303) ￥4,800	CP/M68Kエミュレータ(B6-6302) ￥19,800

PRKニューバリエーション販売開始! PRK10コプロセッサ付/メモリー無し 定価￥72,000

BASICHOUSE BBS TECOSYS NET

TEL 0286-27-1829 / 1200/2400ボー・MNPクラス5/8ビット/バリティ無し/X制御無し
ゲストIDなし(オンラインサインアップ)

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研

本社営業部/マイコンショップ/通販部 宇都宮市竹林町503-1 TEL0286-22-9811 FAX0286-25-3970
大田原営業所/マイコンショップ 大田原市美原1-13-4 TEL0287-23-5352 FAX0286-23-5364

マイコンショップ

BASICHOUSE

お申し込み・お問い合わせは **0286-22-9811(代)**

パソコン専門 O.A.ランド

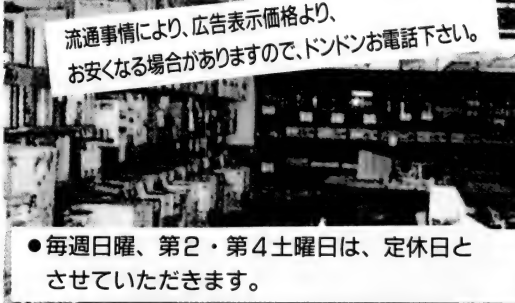
アフターサービス万全
のサポート体制
優良パソコン販売店

- お近くの方は、お立寄り下さい。
専門係員がアドバイスいたします。
- ビジネスソフト、ゲームソフトのこと
ならおまかせ下さい!!

セール期間

◀ '90 7・15 ▶ 8・15

サマーセール!! ドカ〜とプレゼント
OAランド恒例・大お買得セール実施中



●毎週日曜、第2・第4土曜日は、定休日とさせていただきます。

SHARP X68000シリーズセット (お楽しみゲームパック付)

●次代のインテリジェンス= SX-WINDOW搭載!!

X68000 EXPERT II

- CZ-603C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥453,000



X68000 EXPERT II-HD

- CZ-613C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥563,000

OAランド大特価

クレジット	12回	24回
	¥30,200	¥15,900

NEW

OAランド大特価

クレジット	12回	24回
	¥37,400	¥19,700

X68000 PRO II

- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥400,000



X68000 PRO II-HD

- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計 ¥510,000

OAランド大特価

クレジット	12回	24回
	¥26,600	¥14,000

OAランド大特価

クレジット	12回	24回
	¥33,800	¥17,800

X68000 SUPER-HD

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック



X68000 SUPER-HD

- CZ-623C-TN(チタン)
- CZ-613D-TN(チタン)
- MD-2HD 20枚

定価合計 ¥633,000

クレジット	12回	24回
	¥40,600	¥21,400

NEW

OAランド大特価

X-1ターボⅢセット

①セット

- CZ-88CBK...定価¥169,800
- CZ-88DBK...定価¥109,800
- CZ-6ST1B...定価¥5,800
(チルトスタンド)
- MD-2HD 20枚サービス

合計定価¥275,400

特価中TEL下さい

②セット

- CZ-888CBK...定価¥169,800
- CZ-830DBK...定価¥98,000
- CZ-6ST-1B...定価¥5,800
(チルトスタンド)
- MD-2HD 20枚サービス

合計価格¥273,600

特価中TEL下さい



今月の特価品(限定)お早目に!!

★CZ-652C(BK)+CZ-602D(BK)

4セット限り...大特価¥258,000

- SHARP WD-A300(フープロ) 定価¥165,000 特価¥110,000
- SHARP WD-A330(フープロ) 定価¥185,000 特価¥125,000
- SHARP WD-HL30(フープロ) 定価¥198,000 特価¥134,000
- SHARP PW-910(フープロ) 定価¥85,000 特価¥85,000

★CZ-612C(BK)

3セット限り...大特価¥298,000

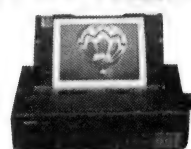
- NEC PC-KD853(アナログCRT) 定価¥50,000
- 三菱XC-1498C(アナログCRT) 定価¥54,800
- SHARP CU-14FD(アナログCRT) 定価¥46,000
- SHARP PA-8500(電子手帳) 定価¥16,000

周辺機器コーナー

プリンターセットコーナー

- CZ-6PVI(カラービデオプリンター) 定価¥198,000 特価¥152,000
- CZ-8PC3(24ドット熱転写カラープリンター) 定価¥65,800 特価¥53,000
- CZ-8PK10(24ピン漢字ドットプリンター・136桁) 定価¥97,800 特価/TEL下さい!
- CZ-8PG1(24ピンカラー漢字ドットプリンター・80桁) 定価¥130,000 特価/TEL下さい!
- CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁) 定価¥160,000 特価/TEL下さい!
- IO-735X(カラーイメージジェットプリンター) 定価¥248,000 特価/TEL下さい!

OAランド特選品!!



■CZ-8PC4(定価¥99,800)

●48ドット熱転写カラー漢字プリンター 特価¥64,800

X68000用ソフトウェア・コーナー

- ①CZ-212BS(BUSINESS)...定価¥68,000 特価¥53,000
- ②CZ-220BS(DATA)...定価¥58,000 特価¥45,000
- ③CZ-215MS(Sampling)...定価¥17,800 特価¥13,800
- ④CZ-21HS(NEW Print Shop)...定価¥10,800 特価¥15,500
- ⑤CZ-227BS(TOP財務会計)...定価¥200,000 特価¥158,000
- ⑥CZ-226BS(CARD)...定価¥229,800 特価¥23,000
- ⑦CZ-223CS(Communication)...定価¥19,800 特価¥115,500
- ⑧CZ-213MS(MUSIC)...定価¥18,800 特価¥14,800
- ⑨CZ-211LS(C compiler)...定価¥39,800 特価¥31,000
- ⑩C-TRACE(キャスト)...定価¥68,000 特価¥52,000
- ⑪EW(イースト)...定価¥38,000 特価¥29,000

X68000用周辺機器コーナー

- CZ-6PU1A...定価¥38,000 特価¥30,000
- CZ-6BM1...定価¥26,800 特価¥21,000
- CZ-6BE1...定価¥88,000 特価¥69,800
- CZ-6VT1...定価¥69,800 TEL下さい!
- CZ-8NS1...定価¥188,000 特価¥149,000
- CZ-6BC1...定価¥79,800 特価¥63,000

●最新ゲームソフト

その他各種ソフト

20%~25%OFF!!

●周辺機器・プリンター

割引販売中!! TEL下さい!

■I-O DATA 増設RAMボード

●1MB増設RAMボード

PIO-6BE1-A

定価 ¥25,000

●2MB増設RAMボード

PIO-6BE2-2M

定価 ¥50,000

●4MB増設RAMボード

PIO-6BE4-4M

定価 ¥88,000



特価¥19,500 特価¥38,500 特価¥67,000

■ハードディスク ■特価品もありますので TEL下さい。

- アイテック ITX-640...特価¥117,000
- アイテック ITX-680...特価¥149,000
- ロジテック LHD-32V...特価¥85,000
- ロジテック LHD-34VE...特価¥90,000
- ロジテック LHD-34V...特価¥104,000
- シャープ CZ-620H...特価¥118,000
- シャープ CZ-64H...特価¥95,000
- アイテム HXD-040...特価¥88,000
- アイテム HXD-042...特価¥95,000
- ICM SR-80...特価¥130,000

中古パソコン

(価格/在庫は変動します。予約は5日以内とします。)

PC-9801RA5	¥338,000	PC-286VS	¥165,000
PC-9801RA2	¥265,000	CZ-600C	¥160,000
PC-9801RX2	¥199,000	CZ-601C	¥170,000
PC-9801EX2	¥190,000	CZ-611C	¥198,000
PC-9801VX21	¥170,000	CZ-652C	¥178,000
PC-9801UX21	¥165,000	CZ-612C	¥210,000
PC-9801VX2	¥160,000	68000用モニター	¥49,000
PC-9801VM21	¥150,000	PC-9801用サウンドボード	¥13,000
PC-9801UV11	¥148,000	PC-88SR.FR	¥50,000
PC-9801LV22	¥160,000	PC-88FH.FA	¥65,000
PC-286VE	¥150,000	400ラインCRT	¥38,000
PC-286US	¥155,000	200ラインCRT	¥10,000

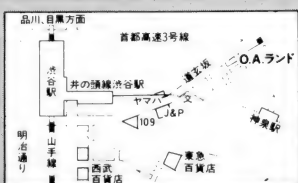
通信販売のご案内

全国通販

■銀行振込で申し込みの方は商品名
及びお客様の住所・氏名・電話番号
をお知らせ下さい。

(振込先)第一勧業銀行 渋谷支店
普通No.1163457 株オーエーランド

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1~60回払で月々5,000円より自由に設定できます。



●下取・買取は電話で見積りしております。責任を持って下取りさせていただきます。

●ご注文、お問合せは... 午前10時から午後7時まで

●商品のお届けは...入金確認後、即日発送致します。

株オーエーランド

〒150 東京都渋谷区円山町20-4 第5日新ビル1F

☎(03)770-8855 FAX (03)770-7080

関東エリアの送料は、1個につき¥1,000です。

★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合わせ下さい。掲載の価格は、6月末現在です。

株式
会社

デンキヤ



営業時間AM11:00~PM7:00 水・木曜定休

セット超特価

△ 68000
PERSONAL WORKSTATION

PRO II・PRO II HD

CZ-653C特価

CZ-663C特価

SUPER HD

CZ-623C特価

CZ-613D特価

(価格はすべて税込みです)

セット超特価

△ 68000
PERSONAL WORKSTATION

EXPERT II・EXPERT II HD

CZ-603C特価

CZ-613C特価

EXPERT PRO

CZ-662C特価

CZ-602C特価

全品メーカー保証 即決クレジットOK

ディスプレイ

プリンタ

周辺機器

ソフト

CZ-604D

特価

CZ-8PC4

特価

CZ-8NJ1

¥1,400

CZ-213MS

¥15,500

CZ-605D

特価

CZ-8PG1

特価

CZ-8NJ2

¥18,540

CZ-223CS

¥15,300

CZ-613D

特価

CZ-8PG2

特価

PIO-6BE1A

¥20,000

CZ-219SS

¥23,100

CU-21HD

特価

IO-735X

特価

PIO-6BE2

¥39,000

CZ-211LS

¥30,800

24時間テレホンサービス

0482-54-3444

お申し込み

TEL.0482-54-3400

FAX.0482-54-3443

埼玉県川口市西川口4-6-4

お支払い

下記取引銀行口座

までお振込み下さい。

三菱銀行西川口支店

(株)デンキヤ ③0258081



クリエイイト特典

- 全商品完全保証書付(メーカー保証)
- 全国無料配達(一部離島の方は有料になります)
- 配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- 中古パソコン高値下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00~PM7:00

当社はX68000の販売認定店です。
どんなことでも安心してご相談ください。

X68000
ビッグ・サマーセール開催中!

即売・即納

X68000 NEW PRO II

- CZ-653C(本体).....¥285,000
- CZ-603D(カラーディスプレイ).....¥84,800
- 好きなゲームソフト1本.....¥7,800
- 定価合計.....¥377,600

クリエイイト特価

均等払い	¥7,680×48回	¥9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- CZ-603C(本体).....¥338,000
- CZ-613D(カラーディスプレイ).....¥135,000
- CZ-8NJ2.....¥23,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥506,600

クリエイイト特価

均等払い	¥9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 EXPERT II HD

- CZ-613C(本体).....¥448,000
- CZ-604D(カラーディスプレイ).....¥94,800
- 好きなゲームソフト1本.....¥9,800
- 定価合計.....¥552,600

クリエイイト特価

均等払い	¥5,920×48回	¥7,400×36回	¥12,100×24回
ボーナス	¥30,000×8回	¥40,000×6回	¥50,000×4回

X68000 SUPER HD

- CZ-623C-TN(本体・キーボード・マウス).....¥498,000
- CZ-613D-TN(カラーディスプレイ).....¥135,000
- CZ-6BP1.....¥79,800
- 定価合計.....¥712,800

クリエイイト特価

均等払い	¥7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000×8回	¥50,000×6回	¥80,000×4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ~!

- CZ-603C.....¥338,000
- CZ-605D.....¥115,000
- MU1.B(MIDIボード&ソフト).....¥39,800
- CM32L.....¥69,000
- グラナダ.....¥8,800
- JOYカード.....¥1,800
- 定価合計.....¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET!

- PRO II CZ653C.....¥285,000
- 0.31CRT CZ603D.....¥84,800
- グラナダ.....¥8,800
- Y'S.....¥8,700
- ポピュラス.....¥9,800
- スーパーハンガオン.....¥8,800
- エージャックス.....¥8,800
- サーク.....¥8,800
- アールタイプ.....¥7,800
- アナログJOYSTIC XE-1AP.....¥13,800
- 定価合計.....¥445,100▶超特価¥353,000

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥69,800	MUSIC PRO	MIDI版	¥28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥18,800
CZ-6BE1A	IMB増設RAMボード	¥38,000	SOUND PRO-68K	サウンドエディタ	¥15,800
CZ-6BE2	2MB増設RAMボード	¥79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥28,800
CZ-8NM3	マウス・トラックボール	¥9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥29,800
BF-68PRO	高性能CRTフィルター	¥13,800	PRO-68K	サイバーノート	¥19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥79,800	PRO-68K	ステーションナリー	¥14,800
CZ-8NT1	トラックボール	¥13,800	Ccompiler PRO-68K	ソフト開発セット	¥39,800
CZ-6BM1	MIDIボード	¥26,800	Human 68K Ver2.0	開発ツールセット	¥9,800
CZ-8NJ2	アナログスティック	¥23,800	PIO-6BE1-A	内蔵1MRAM	¥25,000
CZ-6TU	パソコンチューナ	¥33,100	PIO-6BE2-2M	2MRAM	¥50,000
SX-68M	MIDI I/F	¥19,800	PIO-6BE4-4M	4MRAM	¥88,000
XE-1AP	アナログジョイパッド	¥13,800	MU1-B	MIDI I/F+ソフト	¥39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際にお問合せください。●超特価販売中!

オール15%~20%OFF

総合お問合せ先☎03-486-6541代

パソコン専門ショップ

ソフトクリエイイト 渋谷/横浜

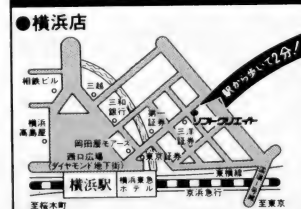
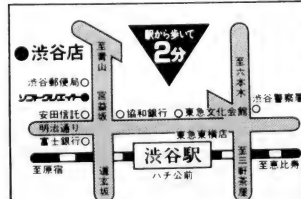
●渋谷店☎03-486-6541(代)

〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店①No.5000340

●横浜店☎045-314-4777(代)

〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店②No.310852

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。



X68000 全機種取り揃え大特価セール

68000 EXPERT/PRO



CZ-602C (本体)
+ CZ-603D (ディスプレイ)
+ SX.WINDOW
大特価 ¥310,000
(このセットに限り、送料+消費税込)
CZ-653C (本体)
+ CZ-602D (ディスプレイ)
大特価 ¥288,000
(このセットに限り、送料+消費税込)

※代金は商品引換着払いでもOKです。

New X68000新発売! (●特価価格は直接お問合せください。)

CZ-603C	定価 ¥338,000	ディスプレイ	
CZ-613C	定価 ¥448,000	CZ-603D	定価 ¥84,800
CZ-623C	定価 ¥498,000	CZ-604D	定価 ¥94,800
CZ-653C	定価 ¥285,000	CZ-605D	定価 ¥115,000
CZ-663C	定価 ¥395,000	CZ-613D	定価 ¥135,000

●新製品も
大特価/お
問合せくだ
さい。

ハガキもOK. New MZプリンタ シャープMZ-1P22

漢字カラー
熱転写プリンタ
C24×24ドット漢字●7色カラー●
漢字30字/秒高速印字●MZ1P
17とフルコンパチ●5KBのバッファ
メモリ付●通信バスコネクタ:MZ2000,
2500, 5500, 6500シリーズ, X1シリ
ーズ, X68000シリーズ他

標準価格 ¥59,800 → **特価 ¥25,000**

パソコンファクス MZ-1V01

“プリンタ・コピー・ファクス”
1台3役のスクレモノ
限定セット販売!

●MZ25セット(インターフェースソフト付)
標準価格合計 ¥342,800 → **¥120,000**

●MZ-1V01 (本体のみ)
標準価格合計 ¥278,000 → **¥98,000**

シャープMZ-1X30 モデムホン

(1×19上位機種)
●300/1200bps全2重通信対応
モデム内蔵 ●音声入出力端子
付 ●ダイヤルパルス/フッシュボ
タン対応 ●フッシュボタン音解析
機能 ●シャープ手帳、CGITT、V25
bis通信手順サポート

標準価格 ¥98,000 → **大特価**

パソコンと専用ワープロをひとつにした16ビット シャープMZ-2861

ワープロソフト「書院28」
MS-DOS V3.1 装備
エミュレーションソフト搭載
定価 ¥328,000 →
大特価 ¥198,000
(ディスプレイは別)

MZ-2861用ソフト(UPシリーズ)

●IP-1251(デスクトップ).....定価 ¥88,000 → 特価 ¥20,000
●IP-1253(クリップ).....定価 ¥77,000 → 特価 ¥20,000
●IP-1254(ランナー).....定価 ¥88,000 → 特価 ¥20,000

シャープMZ-2520

定価 ¥159,800 → **大特価 ¥78,000**

《在庫限り》 PC-E500PJ

定価 ¥28,800 → **大特価**
SHARPポケコン全機種、
Z80ボード他、太平洋工業製品全機種取扱
●PC-E500PJご購入の方に、もちろん
「ポケコンジャーナル特別号」を進呈!

PC-500と各種パソコンをつなぐインターフェースケーブル CE-140T ¥8,800

SHARPラップトップパソコン AX286L-F



定価 ¥428,000 →
特価 ¥238,000

拡張機器他

- シャープ CZ-8GR(286拡張)..... ¥32,000 → ¥12,000
- シャープ CZ-8B3(160ボックス)..... ¥33,800 → ¥28,000
- シャープ CZ-88K3(X1)..... ¥13,800 → ¥11,700
- シャープ CZ-88K4(X1)..... ¥6,800 → ¥5,700
- シャープ CZ-88GR2(X1)..... ¥14,800 → ¥4,000
- シャープ CZ-64H(286拡張)..... 特価
- シャープ CZ-8N12(5.25インチ2HD)..... ¥23,800 → ¥18,500
- シャープ MZ-1E11..... ¥38,000 → ¥25,000
- シャープ CZ-811(286拡張)..... ¥8,500 → ¥1,000
- シャープ MZ-1U08..... ¥25,000 → ¥12,000
- シャープ MZ-1U03..... ¥35,000 → ¥15,000
- シャープ MZ-1R22(286拡張)..... ¥21,800 → ¥13,000
- シャープ MZ-1R12(286拡張)..... ¥35,000 → ¥8,000
- シャープ MZ-1E29(MZ)..... ¥17,800 → ¥9,800
- シャープ MZ-1E30..... ¥25,000 → ¥22,500
- シャープ MZ-1U09(2500)..... ¥9,000 → ¥7,200
- シャープ MZ-1M03(5500)..... ¥69,000 → ¥35,000
- シャープ MZ-8B04(2000)..... ¥18,000 → ¥8,000
- シャープ MZ-8B10(2000)..... ¥45,000 → ¥18,000
- シャープ MZ-1R11(5500)..... ¥80,000 → ¥30,000
- シャープ MZ-1R24(1500)..... ¥22,000 → ¥6,000
- シャープ MZ-1R26A(2500)..... ¥13,000 → ¥12,800
- シャープ MZ-1R27A(2500)..... ¥13,000 → ¥10,000
- シャープ MZ-1R28A(2500)..... ¥13,000 → ¥10,000
- シャープ MZ-1T03(1500)..... ¥32,000 → ¥10,000
- シャープ MZ-1T02(2200)..... ¥19,800 → ¥8,500
- シャープ MZ-1X09(1500)..... ¥12,000 → ¥8,500
- シャープ MZ-1X29..... ¥13,800 → ¥11,000
- テリスシステム-ZSE(5500拡張)..... ¥428,800 → ¥38,500
- シャープ CZ-65T1(286拡張)..... 特価 ¥3,500
- シャープ CZ-8B2(160拡張)..... ¥29,800 → ¥25,300
- シャープ CZ-8B2(160拡張)..... ¥19,800 → ¥16,800
- シャープ X1 MZ用マウス..... 特価 ¥4,800
- シャープ X1用ジョイカード..... ¥1,500
- シャープ MZ-5500キーボード..... ¥8,000
- シャープ 2000/2200キーボード..... ¥8,000
- シャープ MZ-1E08..... ¥9,000 → ¥8,000
- シャープ CZ-6B11(286拡張)..... ¥26,800 → ¥23,000
- シャープ CZ-4B1(286拡張)..... ¥35,000 → ¥29,500
- シャープ CZ-4B1E(X68000拡張)..... ¥38,000 → ¥23,800
- アイトックPO-882M(160拡張)..... ¥25,000 → ¥21,500
- アイトックPO-882M(160拡張)..... ¥50,000 → ¥42,500
- アイトックPO-884M(160拡張)..... ¥88,000 → ¥74,500

(MZ-2861)

- シャープ MZ1R35(286拡張)..... ¥55,000 → ¥19,000
- シャープ MZ1R36(286拡張)..... ¥45,000 → ¥15,000
- シャープ SS-C28M(286拡張)..... ¥49,800 → ¥10,000

プリンター

- シャープ CZ-8PG1(4色・グレー)..... ¥99,800 → 大特価
- シャープ CZ-8PG2..... ¥130,000 → ¥100,000
- シャープ CZ-8PG2..... ¥160,000 → ¥130,000
- シャープ MZ-1P27..... ¥268,000 → ¥214,400
- シャープ MZ-1P28..... ¥148,000 → ¥118,400
- シャープ MZ-1P29..... ¥168,000 → ¥134,400
- シャープ MZ-6P18..... ¥60,000 → ¥35,000
- シャープ MZ-6P27..... ¥58,000 → ¥39,800
- シャープ MZ-6P29..... ¥50,000 → ¥37,500

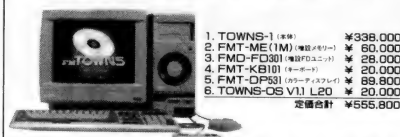
フロッピーディスク

- シャープ CZ-501H(286拡張)..... ¥258,000 → ¥60,000
- シャープ CZ-503F..... ¥49,800 → ¥30,000
- シャープ CZ-53F..... ¥19,800 → ¥9,800
- シャープ CZ-300F(CZ-3PCM付)..... ¥13,000

1BIT

アイビット電子株式会社

FM TOWNS お買い得セット



1. TOWNS-1 (本体)..... ¥338,000
2. FMT-ME1M (ハードディスク)..... ¥60,000
3. FMD-FD3M (ハードディスク)..... ¥28,000
4. FMT-KB10 (キーボード)..... ¥20,000
5. FMT-CP3M (ソフトウェアセット)..... ¥88,800
6. TOWNS-OS V1.1.280..... ¥20,000
定価合計..... ¥555,800

大特価/¥285,000

MZ2500下取り/MZ2500からMZ2861 (定価 ¥328,000)に買い替え下取後 特価 ¥165,000
CZ600C下取り/CZ600CからCZ623(X68000 SUPER)に買い替え下取後 特価 ¥300,000

ハードディスク

- アイテックIT-X640..... ¥158,000 → ¥128,000
- アイテックIT-X68..... ¥198,000 → ¥158,000

ディスプレイ

- シャープ MZ-1D17(286拡張)..... ¥124,000 → ¥63,000
- シャープ MZ-1D27..... ¥120,000 → ¥79,800

ソフト

- (X68000用)
- CZ-230AS ニューゼaland..... ¥8,800 → ¥7,040
- CZ-230AS FULL THRTILE..... ¥8,800 → ¥7,040
- CZ-232AS PACMANIA..... ¥7,800 → ¥6,250
- CZ-223AS ARKANOID..... ¥7,800 → ¥6,250
- POPULOUS..... ¥9,800 → ¥7,850
- CZ-239AS THUNDARBLADE..... ¥9,500 → ¥8,000
- CZ-259S X68000XWindow..... 特価
- (MZ-2500用)
- IP-1215 COBOL..... ¥13,800 → ¥11,700
- DANGER BOX..... ¥5,800 → ¥2,000
- EXTRA HYPER DISK MONITOR..... ¥10,000 → ¥8,500
- EXTRA HYPER DISK MONITOR..... ¥14,000 → ¥12,000
- FILE UTILITY CUT-25F..... ¥6,800 → ¥6,000
- FREE CALL..... ¥6,800 → ¥1,000
- G-EDIT2500..... ¥8,800 → ¥7,000
- H.S.コントローラー..... ¥9,800 → ¥8,500
- HuCAI日本語..... ¥45,000 → ¥15,000
- カレイドスコープ..... ¥9,800 → ¥3,000
- カレイドスコープ2..... ¥5,800 → ¥1,000
- ザ・ブラックオニキス..... ¥7,800 → ¥3,000
- スーパー修理屋さん..... ¥12,000 → ¥10,200
- ムーンチャルド..... ¥7,800 → ¥3,000
- 英雄伝説ザ・サガ..... ¥9,800 → ¥2,000
- 五目並べ..... ¥4,800 → ¥2,000
- 接線隊2弾..... ¥7,800 → ¥2,000
- プリントSHOP..... ¥9,800 → ¥8,500
- プリントSHOPライブラリー1..... ¥4,500 → ¥3,800
- プリントSHOPライブラリー2..... ¥4,500 → ¥3,800

(X1用)

- 日本語ワープロ将軍X1..... ¥34,800 → ¥29,000
- 日本語ワープロ侍X1..... ¥19,800 → ¥16,800
- CZ-8W51 X1ディスクBASIC..... ¥9,800 → ¥3,500
- 3CP/M X1 3'CP/M..... ¥16,800 → ¥5,000
- CZ-8B3 X1第2水準ROM..... ¥13,800 → ¥11,700
- CZ-12B5F X1 CP/M..... ¥13,800 → ¥11,700
- CZ-1305F X1 CP/M..... ¥14,800 → ¥12,500
- CZ-116F X1 C..... ¥13,800 → ¥11,700
- CZ-117S X1 LOGO..... ¥18,800 → ¥13,200
- CZ-118F X1 COBOL..... ¥13,800 → ¥11,700
- CZ-126F X1 APL..... ¥13,800 → ¥11,700
- CZ-115F X1 FORTRAN..... ¥11,700

(MZ-5500, 6500SOFT)

- MZ-22013 (MZ-5500MSDOS)
- MZ-22014 (MZ-5500TODAY)
- MZ-22023 (MZ-5500GW BASIC)
- MZ-22028 (MZ-5500GW BASIC)
- MZ-22025 (MZ-5500ワープロ)
- MZ-22029 (MZ-5500TODAY)

本体 ●シャープ CZ-820, 822, 880, 881, MZ-3500, 2520, 2861, 2200, X68000, CZ-612, 662, 602, 652 ●富士通 FM-77AV.1, 77AV.2, 77AV.20, 77AV.40 ●NEC PC-9801IN ●東芝 J3100SS

SHARP プリントアップパソコン
All in Note 新発売! 入荷
AX286N-H2 定価 ¥398,000

《全商品新品完全保証付》■シャープポケコン全商品販売中。カタログ、特価表ご請求ください。(〒72)

0426-45-3001~3
FAX.0426-44-6002

●営業時間/10:00~19:00 ●電話受付/20:00迄可 ●定休日/日曜日(祭日営業)

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

●本誌発売時には上記価格よりさらに求めやすい価格に変更されている場合があります。●一部を除き、上記商品価格には消費税は含まれておりません。その商品に対し別途3%の消費税がかかりますのでご了承ください。

上記の広告商品はすべて店頭販売もしております

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でも申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの際はご容赦下さい。

全通販 国信売
北海道から沖縄まで
富士銀行八王子支店 (普) 1752505

PERSONAL COMPUTER MAGAZINE
PC-9801活用誌

Oh!PC

パソコン
テスト
マガジン

PG
MAGAZINE
JAPANESE EDITION

THE
COMPUTER
コンピュータ時代を読む トレンド・マガジン

C
MAGAZINE

ソフトバンク
の
雑誌が勢揃い

月刊・コンピュータ技術者必携
第2種・第1種・特種受検

情報処理試験

Oh!FM

Oh!△

BEEP! POWERFUL MEGA-MAGAZINE
MEGADRIIVE
メガドライブ

《広告の半ページ》 ああ、また暑い夏がくる。♪ヘナヘナヘーナーヘエナ〜 ちからがめーけーるー

月刊 電脳倶楽部 90年8月号 (Vol.27)

7月18日発送

2HDディスクに入ったX68000のための雑誌だっ!

もしかして

CW送受信練習プログラム

さらにもしかして

X-BASIC配列サーチ・ソート関数

さらには

CZ-8PC2用
カラーハードコピープログラム

そして

しりとりPRO-68K

とどめはPDDで

軽犯罪法・日米安保条約

その他、便利なツール、PDD、ビープ音、読み物などを満載!

(なお、内容は一部変更されることがあります。ご了承下さい)

編集長祝一平からの御挨拶「どーもどーも、暑いですねえ。今日なんかアイスキャンデーを五本も食べてしまいました。五本といえば龍角散」

満開製作所 電脳倶楽部
編集部

〒171 東京都豊島区要町1-19-3 いさみビル4F
TEL.(03)554-9282/FAX.(03)554-3856

販売方法は通信販売のみです。お申し込みの方法は左記の住所へ現金書留で
定期購読 6ヶ月分 6,000円 (消費税込・郵送料サービス)

●7月18日以降に受け付けた分は、原則としてVol.27から発送します。新たに購読を希望される方は、「新規」と御明記下さい。

●郵便振替を御利用の場合は口座番号「東京5-362847 満開製作所」でお願いいたします。製品の性格上、返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返します。

(ご注意：バックナンバーの受け付けは、定期購読の方に限らせていただきます)

X1 エミュレータ

好評発売中

定価¥9,800



X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3~5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1 エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリーイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

X1ディスク ↔ X68000 Human68k (5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC: CP/M ↔ X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。
※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。



X1 エミュレータ Q&A

- Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?
A. 専用のケーブルが付属しますのでその必要はありません。
- Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?
A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。
- Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?
A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。
- Q. Turbo用のソフトは動きますか?
A. X1用のみでTurbo専用のソフトは動きません。
- Q. ゲームは動きますか?
A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。
- * タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
* 一部サポートしていない機能があります。
- X1エミュレータ通信販売** 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

*この商品価格には消費税は含まれておりません。

*CP/Mはデジタルリサーチ社の商標です。

文中のソフトウェアは各社の商標です。

*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社 **アクセス** 〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
TEL 03 (233) 0200(代) FAX 03 (291) 7019

資料請求券
お申し込み
8月号

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE SIG探訪PART 2

ライターズポート/書くネット!

(ジャンプコード:WRITERS)



TOKYO

SIGのサブオペ(サブオペレーター)が東京にいるのは心強い!ライターにとって気になる東京の、新鮮な情報がどんどん入ってきます。



OSAKA

ちょっと声をかければ、ワツと集まり、オフラインミーティングが始まる。2~3人でお酒をのみながら議論することもあれば、ワイワイ騒ぎながら異業種交流したりもしています。



ライターがどうして畑仕事を
してるかって??
ライターももえる!
これぞまさに
ライターの畑作法!
なんのこっちゃ

「書きこむ」人はみなライター! プラスワンをめざす異業種交流SIG.

東京・大阪をまたにかけ、MSG(メッセージ)が飛び交ってゆく。ここは電腦ライター御用達、書き屋の港「ライターズポート/書くネット!」プランナーやらコピーライター・シナリオライターが、自由気ままに好き勝手の活動をしているから、活動内容は多種多様。各種広告・映像・出版物の批評/感想など朝メシ前で、電腦ライター必須の技術(?)、ブラインドタッチなら、自作のタイピング練習ソフトを開発するという行動力!その他にも自作の芝居を公演したり、なぜだか畑仕事(!)にまで手を出してしまいます。テーマはつねにプラスワン。「書きこむ」人はみなライターと定義して、異業種交流を基本にパソコン通信と現実の活動とリンクさせているのです。書くこと、演劇、畑仕事、広告etc.興味のある方はこぞっておいでください。

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品の会員割引もあるONLINE SHOPPING.
- ★J&Pだから強い!!パソコン情報ははじめとする役に立つDATA BASE.
- ★みんなでしゃべりオンライントーク(CHAT機能).
- ★地域別・テーマ別ボードで充実のBBS(電子掲示板).
- ★ビジュアルデータもばっちり送受信できるX-MODEM.

J&P HOT LINEへのご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

〒556 大阪市浪速区日本橋5-16-5 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店 東京都渋谷区道玄坂2丁目28番4号 ☎(03) 496-4141
町田店 東京都町田市森野1丁目39番16号 ☎(0427) 23-1313
八王子店 東京都八王子市旭町1番1号八王子そごうF ☎(0426) 26-4141
立川店 東京都立川市幸町4-39-1 ☎(0425) 36-4141
本厚木店 厚木市中町3-4-3 ☎(0462) 25-1548
富山店 富山市桜町2-1-10 ☎(0764) 32-3133
金沢店 金沢市入江2-63 ☎(0762) 91-1130
寺地店 金沢市寺地2-3 ☎(0762) 47-2524
大須店 名古屋市中区大須4丁目2-48 ☎(052) 262-1141

テクノランド 大阪市浪速区日本橋5丁目6番7号 ☎(06) 634-1211
メディアラント 大阪市浪速区日本橋5丁目8番26号 ☎(06) 634-1511
コスモランド 大阪市浪速区難波中2丁目1番17号 ☎(06) 634-3111
U.S.LAND 大阪市浪速区日本橋4丁目9番15号 ☎(06) 634-1411
ビジネスランド 大阪市北区梅田1-1-3大阪駅前第3ビル82 ☎(06) 348-1881
梅田店 大阪市北区小松原町1-10 ☎(06) 362-1141
高槻店 高槻市高槻町11番16号 ☎(0726) 85-1212
くすは店 枚方市楠葉花園町15番2号 ☎(0720) 56-8181
千里中央店 豊中市新千里東町1-3SENCHU PAL2番街4 ☎(06) 834-4141
摂津富田店 高槻市大畑町24-10 ☎(0726) 93-7521
寝屋川店 寝屋川市緑町4-20 ☎(0720) 34-1166

藤井寺店 藤井寺市岡2丁目1番33号 ☎(0729) 38-2111
岸和田店 岸和田市土生町2451-3 ☎(0724) 37-1021
さんみやばん 神戸市中央区八幡通3-2-16 ☎(078) 231-2111
西宮店 兵庫県西宮市河原町5-11 ☎(0798) 71-1171
姫路店 姫路市東延未1丁目1番住友生命姫路南ビルF ☎(0792) 22-1221
京都寺町店 京都市下京区寺町通仏光寺下ル東通之町545 ☎(075) 341-3571
京都近鉄店 京都市下京区烏丸通七条下ル東通小路702 ☎(075) 341-5769
和歌山店 和歌山市元寺町4丁目4番地 ☎(0734) 28-1441
奈良1ばん館 奈良市三条町478-1 ☎(0742) 27-1111
郡山インター店 大和郡山市横田693-1 ☎(07435) 9-2221
熊本店 熊本市手取本町4-12 ☎(096) 359-7800

ADVANCED TURBO

先駆の“Z”アビリティがパソコンクリエイターを魅了する。



AV パソコンテレビ **turbo Z III**

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK 標準価格 169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK 標準価格 92,200円(税別)
テレビスタンド	CZ-6ST1-B 標準価格 5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロップ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに広げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV, RGB, HALF, CDOWN, CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート ●1Mバイトの5インチフロッピーディスクドライブ2基搭載 ●JIS第1/第2水準標準漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能 ●ニューデザインのマウス標準装備 ●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計 ●プリンタ、RS-232Cなど豊富なインターフェイスを装備 ●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。